



UNIVERSIDAD NACIONAL DE INGENIERÍA
Recinto Universitario Simón Bolívar
Facultad de Electrotecnia y Computación

TRABAJO MONOGRÁFICO

“Sistema de información web, para la gestión de la Vivienda Estudiantil y los becados residentes de la Universidad Nacional de Ingeniería”

PARA OPTAR AL TÍTULO DE
INGENIERO EN COMPUTACIÓN

Elaborado por:

Br. Francel William Palacios Martínez

Br. Kenneth David Gaitán Evanks

TUTOR:

MSc. Norman René Trujillo Zapata

Managua, Nicaragua

2016

DEDICATORIA

Este trabajo monográfico se lo dedicamos primeramente a Dios, por permitirnos culminar con éxito esta etapa de nuestras vidas y darnos sabiduría, que, a pesar de todas las dificultades, días ocupados, Dios nos ayudó a superarlas cada una de ellas.

A nuestros padres, sin su apoyo este trabajo monográfico no hubiera sido posible, este logro es un reconocimiento para ellos, por sembrar recursos y tiempo en nosotros.

A cada uno de nuestros seres queridos, que de alguna u otra forma nos ayudaron con sus palabras de aliento y apoyo económico.

AGRADECIMIENTOS

Gracias a Dios y a nuestros padres por la ayuda brindada en cada situación.

Al MSc. Norman Trujillo por acompañarnos en todo el proceso de elaboración de este trabajo monográfico y brindar una excelente tutoría.

A la Esp. Angélica María Saborío Pérez por haber revisado el documento y brindarnos algo que jamás podremos pagar, su valioso tiempo.

A la Universidad Nacional de Ingeniería por brindarnos la oportunidad de estudiar y formarnos como profesionales.

Gracias a cada profesor que nos impartió clases compartiendo sus experiencias y conocimientos a lo largo la carrera de ingeniería en computación.

RESUMEN

La búsqueda de optimización en los procesos administrativos permanece como un tema de mucho interés en las reuniones gerenciales de grandes empresas u organizaciones, ya que siempre se está buscando una forma de mejorarlos.

Actualmente los sistemas de información (SI) tienen un papel importante en las organizaciones, ya que ayuda en los procesos administrativos, optimizando recursos como el tiempo y dinero en actividades diarias. En la mayoría de las empresas e instituciones cuentan con SI que ayudan en la contabilidad, control de clientes, pedidos de comida en línea, etc. La gran ventaja de estos sistemas es que se adaptan a la necesidad de cada giro de negocio, brindando así, soluciones que satisfagan necesidades particulares y exclusivas de cada cliente.

El presente proyecto surge como respuesta a la necesidad de dotar de un sistema de información web que ayude al encargado de la Vivienda Estudiantil de la Universidad Nacional de Ingeniería. De esta manera se agilizarán funciones del encargado de la misma, dado que el sistema facilitará la gestión de los procesos administrativos, de la Vivienda así como los servicios ofrecidos a los becados residentes.

El sistema de información web se concluyó satisfactoriamente. El encargado reviso cada funcionalidad en el sistema web, examinó los datos que se mostraban y dio el visto bueno, y procederá a hacer las gestiones junto con la Dirección de Bienestar Estudiantil para implementar el sistema en la Vivienda Estudiantil.

ÍNDICE

1. Introducción	1
2. Objetivos	3
2.1 Objetivo General	3
2.2 Objetivos Específicos.....	3
3. Justificación	4
4. Marco Teórico	6
4.1 Sistema de Información	6
4.2 Sistema Web	8
4.3 Servidores.....	8
4.3.1 Servidor Web	10
4.3.2 Servidor de base de datos	10
4.3.3 Microsoft SQL Server	12
4.4 Protocolo de Transferencia de Hipertexto o HTTP	13
4.5 Internet.....	14
4.6 Base de Datos	15
4.7 Framework.....	16
4.8 ASP.NET	18
4.9 Lenguaje Unificado de Modelado o UML	19
4.10 Ingeniería de Software.....	22
4.11 Proceso del software	23
4.12 Metodología de desarrollo ágil SCRUM.....	25
5. Desarrollo de la solución propuesta	29
5.1 Tipo de Investigación	29
5.2 Diseño de Instrumentos de medición.....	29
5.3 Delimitación geográfica.....	33
6. Desarrollo del Sistema de información web	34
6.1 Estudio de factibilidad	34
6.2 Metodología de Desarrollo SCRUM.....	37
6.2.1 Sprint backlog y sprint planning meeting	38

6.2.2. Daily SCRUM.....	85
6.2.3. Sprint Review	85
7. Conclusiones y recomendaciones	86
7.1 Conclusiones	86
7.2 Recomendaciones	87
8. Bibliografía	88
9. Glosario.....	89
10. Anexos.....	92
10.1 Anexo 1: Carta de solicitud de sistema web para la gestión de la Vivienda Estudiantil.....	92
10.2 Anexo 2: Estudio de factibilidad	93
10.3 Estudio de requerimientos	96
10.4 Anexo 3: Factibilidad técnica	101
10.5 Anexo 4: Factibilidad Operativa	110
10.6 Anexo 5: Factibilidad de cronograma.....	111
10.7 Anexo 6: Factibilidad Legal	115
10.8 Anexo 7: Factibilidad económica	117
10.9 Anexo 8: Entrevista de levantado de requerimientos.....	129
10.10 Anexo 9: Documento de proceso de observación	130
10.11 Anexo 10: Lista de ISO, Usadas principales.....	133
10.12 Anexo 11: Plantillas de caso de uso	135
10.13 Anexo 12: Diagramas lenguaje unificado de modelado (UML) para el diseño lógico del sistema de información web para la Vivienda Estudiantil ..	145
10.14 Anexo 13: Load y perfomance Test	161
10.15 Anexo 14: Test validación de información de salida y entrada	163

Índice de Tablas:

Tabla 1: Descripción de elementos de medición (Elaboración propia)	32
Tabla 2: Equipo de Desarrollo del sistema de información web	38
Tabla 3: Descripción de perfil administrador del sistema.....	96
Tabla 4: Descripción de perfil administrador Vivienda Estudiantil	97
Tabla 5: Descripción de perfil de estudiante.....	97
Tabla 6: Detalle de interfaces de usuario	99
Tabla 7: Detalle de interfaz de menú.....	99
Tabla 8: Descripción de diseño responsivo	100
Tabla 9. Propuesta 1	117

Índice de Figuras

Figura 1 Diagrama conceptual de un sistema de información (Pressman, 2010).	7
Figura 2: Diagrama proceso SCRUM (Elaboración propia)	28
Figura 3: Ubicación geográfica Universidad Nacional De Ingeniería (fuente Google Maps)	33
Figura 4: Listado de Sprints	39
Figura 5: Interfaz del sistema, teléfono inteligente Nexus 5x con resolución de pantalla de 412 x 732 pixeles (Elaboración propia)	49
Figura 6: Interfaz del sistema, Ipad con resolución de pantalla de 1024 x 768 pixeles.	49
Figura 7: Interfaz del sistema desde una PC con resolución de pantalla de 1366 x 768 pixeles.	50
Figura 8: Flujo diseño de base de datos (Elaboración propia)	53
Figura 9: Muestra diagrama Entidad-Relación (ERD)	56
Figura 10: Interfaz de Formulario para Registro de Horarios de Alimentación	58
Figura 11: Interfaz de Formulario para Listar Horarios de Alimentación	59
Figura 12: Interfaz de Formulario con diseño adaptativo para Listar Horarios de Alimentación (vista desde un teléfono Nexus 6 P)	59
Figura 13: Navegabilidad Jerárquica del sistema	61
Figura 14: Arquitectura de Software (Elaboración propia)	65
Figura 15: Solución de la aplicación en .Net (Elaboración propia)	66
Figura 16: Arquitectura de Capas y DDD Aplicada a la solución del Sistema (Elaboración propia)	66
Figura 17: Estructura del sistema de información web para la vivienda estudiantil (Elaboración propia)	67
Figura 18: Diagrama de flujo Modulo de Inventario	69
Figura 19: Diagrama de flujo módulo de Alimentación	70
Figura 20: Diagrama de flujo módulo de creación de actividad social	71
Figura 21: Solución en visual Studio del sistema de información web para la gestión de la vivienda estudiantil	83
Figura 22: Archivos creados para acceso a datos por Entity FrameWork	83
Figura 23: Contenido del archivo app.config	84
Figura 24: Codificación de los procesos administrativos de la vivienda estudiantil	84
Figura 25: Jerarquía de perfiles de usuario (Elaboración propia)	98
Figura 26: Cronograma de proyecto Sistema de información web	113
Figura 27: Línea de tiempo de ejecución del proyecto	114
Figura 28 Resumen de Duración de Cronograma	114
Figura 29: Resultados load test	161

Índice de diagramas

Diagrama 1: Módulo de alimentación	72
Diagrama 2: Módulo de encuestas	72
Diagrama 3: Módulo Inventario.....	73
Diagrama 4: Proceso Registro de enseres / Inventario Nuevo	74
Diagrama 5: Asignación de artículos	75
Diagrama 6: Proceso registro de Universidad o comedor	76
Diagrama 7: Validar entrega de comidas a estudiantes mediante Número De carnet.....	76
Diagrama 8: Proceso Generar rol de aseo	77
Diagrama 9: Proceso de asignación de artículos de inventario	78
Diagrama 10: Proceso de bajas en Inventario.....	78
Diagrama 11: Proceso de configuración de entrega de comidas	79
Diagrama 12: Proceso de validación de entrega de comida a estudiante	79
Diagrama 13: Proceso de asignación de actividad social por estudiante	80
Diagrama 14: Proceso Creación y activación de encuestas.....	80
Diagrama 15: Proceso Contestar encuesta	81
Diagrama 16: Contexto de sistema de información Web para la vivienda estudiantil	82

1. Introducción

El consejo universitario de la UNI, en el año 1991, mediante el acuerdo No. 3, crea la Dirección de Atención Estudiantil, la cual se encarga de los departamentos de Cultura, Deportes y el programa de becas, hasta la actualidad la Dirección de Atención Estudiantil, por su naturaleza de ofrecerles servicios de calidad a los estudiantes, ha creado diversas áreas interdisciplinarias, entre las que se encuentran: Acción social universitaria, Servicio social, Clínica médica estudiantil y el Programa psicopedagógico.

La Dirección de Bienestar Estudiantil (DBE), es una dependencia adscrita a la rectoría, para dar respuestas a las diferentes problemáticas académicas, sociales, deportivas y culturales de los estudiantes durante su formación profesional, a través de programas, proyectos y actividades que los benefician.

Como parte del programa de becas que la DBE ofrece, se encuentra la beca residencial, que se otorga a estudiantes que provienen de los departamentos o zonas de difícil acceso de la capital. La beca contempla alojamiento, alimentación, estipendio monetario, enseres personales, servicios médicos de emergencia y exoneración de matrícula.

En el presente escrito se documenta todo el proceso que se llevó a cabo en la realización del trabajo de desarrollo del sistema de información web. Se incluyen los objetivos que se plantearon como punto de partida, el marco teórico donde se describe la base conceptual de la investigación, las tecnologías y términos involucrados.

Es importante mencionar que en el documento se detalla cómo se desarrolló el sistema de información aplicando la metodología SCRUM, muy usada en la planeación y desarrollo de proyectos informáticos. Así mismo, se describen las técnicas empleadas, como las entrevistas con el cliente para el levantamiento de requerimientos, el análisis de los requerimientos y el diseño del software con base en las necesidades del cliente.

De igual manera, se muestran los resultados obtenidos del análisis de factibilidad sobre la viabilidad del proyecto en diversos aspectos, como son: económico, operativo, técnico y también con base en el cronograma elaborado con la estimación de los tiempos. Al final, se describen las conclusiones de cada aspecto antes mencionado, con una serie de recomendaciones para mejorar el proyecto.

2. Objetivos

A continuación, se presentan los objetivos que guiaron este proceso de desarrollo del sistema de información web.

2.1 Objetivo General

Desarrollar un sistema de información web que facilite la gestión de la Vivienda Estudiantil y los servicios ofrecidos a los becados residentes, de la Universidad Nacional de Ingeniería.

2.2 Objetivos Específicos

- Analizar los procesos de gestión administrativa realizados en la Vivienda Estudiantil, determinando así los requerimientos necesarios para el diseño del sistema de información web.
- Diseñar las interfaces, arquitectura, algoritmos, bases de datos y flujos de datos del sistema, empleando técnicas y estándares de ingeniería de software.
- Codificar los procesos y funciones con base en el diseño realizado, empleando técnicas algorítmicas pertinentes al lenguaje de programación, que permitan la eficiencia en la consulta, procesamiento y escritura de los datos.
- Validar el sistema de información web para la gestión de la Vivienda Estudiantil, realizando pruebas que garanticen el correcto funcionamiento y eficiencia del mismo, así como su coherencia de datos con respecto a los requisitos del análisis y diseño.

3. Justificación

La Vivienda Estudiantil de la UNI, como encargada de ejecutar todas las acciones relacionadas con la beca residencial debe cumplir con los estudiantes residentes los procesos de inventario, entrega de alimentos, estipendios y enseres de limpieza. De igual manera organizan actividades de servicio social, elaboración del rol de aseo, encuestas de seguimiento, que miden el nivel de satisfacción estudiantil. Cada proceso es dirigido, actualmente por el encargado o administrador, Licenciado Humberto Pérez.

Considerando que solamente es una persona la encargada de llevar a cabo los procesos antes mencionados, hay márgenes de error difíciles de evitar. Se han presentado problemas al momento de la distribución de comida, debido a que las entregas se hacen en el comedor, y en ocasiones por descuido del personal de la cocina se entregan más de dos servicios de comida al mismo estudiante, o bien se presentan casos donde la misma persona aprovecha y retira dos veces servicios de alimentación, ya que no hay una manera efectiva de controlar las entregas.

Las órdenes de comida que se entregan por duplicado, generan pérdidas en dinero a la administración de la vivienda de residentes becados, y a su vez priva a otros de poder recibir el beneficio.

El proceso de control de inventario es elaborado a mano por el administrador de la Vivienda Estudiantil, esta información debe tener una mejor manera de procesar, ya que es información administrativa que refleja gastos en productos de limpieza, utensilios de aseo como lampazos, escobas, muebles, productos consumibles, entre otros.

Otra de las actividades de gran importancia realizadas por el administrador de la vivienda de residentes es organizar jornadas de servicio social con los internos becados. La misma consiste en hacer grupos y asignar un trabajo para su realización, pueden ser jornadas de limpieza, o una asignación especial que manden de la Dirección de Bienestar Estudiantil. Este proceso se dificulta cuando el administrador de la vivienda no puede reunir a todos los participantes por falta de comunicación o bien es imposible controlar o recordar quién participo, ya que es complicado manejar listas de asistencias exactas, también las planificaciones son en ciertas ocasiones informales.

La elaboración del rol de aseo que involucra a todos los estudiantes se vuelve compleja, ya que el administrador de la vivienda debe verificar que el mismo esté correctamente distribuido con todos los internos de la vivienda y en turnos equitativos.

En el caso de continuar la situación de esta manera, los procesos antes mencionados no mejorarían y continuarían algunos fallos en cada semestre, que a largo plazo son pérdidas económicas para la Vivienda Estudiantil. Además se haría mayor uso de recursos como tiempo al momento de hacer una gestión, como la de hacer la invitación a las actividades sociales o rol de aseo.

(Ver Anexo 1: Carta de solicitud de sistema por parte del director de bienestar estudiantil)

4. Marco Teórico

Para el desarrollo de este documento se emplearon las siguientes definiciones:

4.1 Sistema de Información

James A. O'Brien (2006) menciona que un sistema de información (SI) puede ser cualquier combinación organizada de personas, hardware, software, redes de comunicación y recursos de información que almacene, recupere, transforme y disemine datos en una organización. Las personas han confiado en los SI para comunicarse entre sí mediante una variedad de dispositivos físicos (hardware), instrucciones y procedimientos de procesamiento de datos (software), canales de comunicación (redes) y datos almacenados desde los inicios de la civilización.

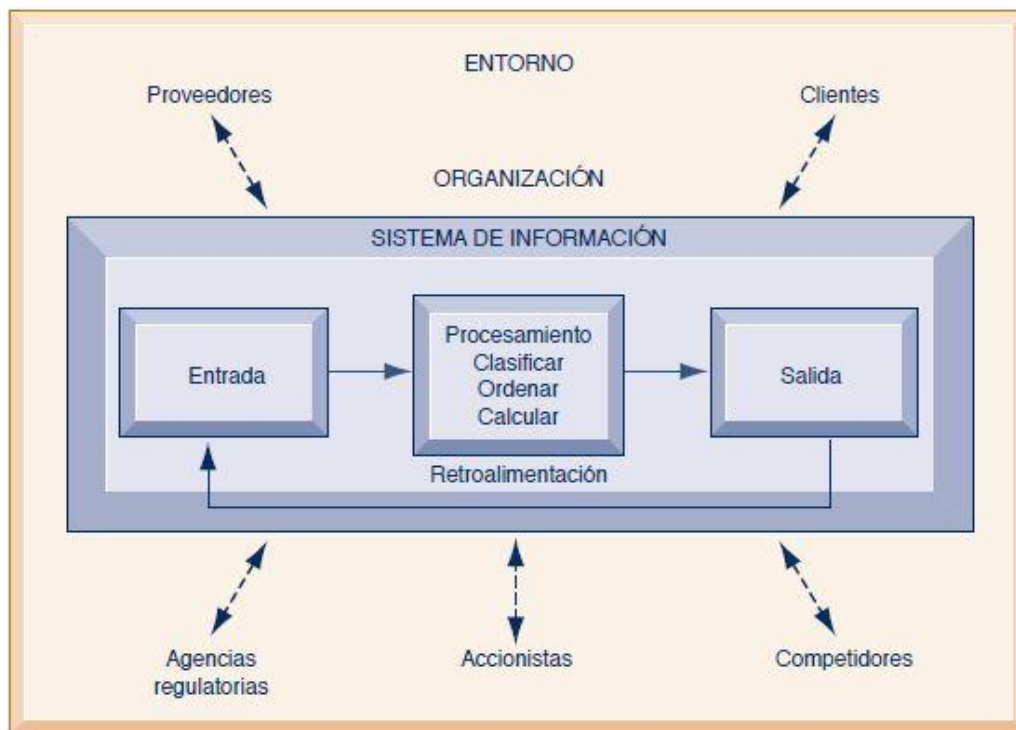
Existen tres funciones vitales que los sistemas de información pueden desempeñar para una empresa:

- Apoyo a sus procesos y operaciones de negocio.
- Apoyo a sus empleados y directivos en la toma de decisiones.
- Apoyo a sus estrategias para conseguir una ventaja competitiva.

Kenneth C. Laudon (2012) comenta que se puede plantear la definición técnica de un SI como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización, beneficiando a los directores, gerentes o jefes.

Hay tres actividades en un sistema de información (SI) que producen los datos necesarios para que las organizaciones tomen decisiones, controlen las

operaciones, analicen problemas y creen nuevos productos o servicios. Estas actividades son: entrada, procesamiento y salida (**Ver Figura 1: Diagrama conceptual de un sistema de información**). La entrada captura o recolecta los datos en crudo desde el interior de la organización o a través de su entorno externo. El procesamiento convierte esta entrada en bruto en un formato significativo. La salida transfiere la información procesada a las personas que harán uso de ella, o a las actividades para las que se utilizará. Los SI también requieren retroalimentación: la salida que se devuelve a los miembros apropiados de la organización para ayudarles a evaluar o corregir la etapa de entrada.



Un sistema de información contiene datos sobre una organización y el entorno que la rodea. Tres actividades básicas (entrada, procesamiento y salida) producen la información que necesitan las empresas. La retroalimentación es la salida que se devuelve a las personas o actividades apropiadas en la organización para evaluar y refinar la entrada. Los actores ambientales, como clientes, proveedores, competidores, accionistas y agencias regulatorias, interactúan con la organización y sus sistemas de información.

Figura 1 Diagrama conceptual de un sistema de información (Pressman, 2010)

4.2 Sistema Web

Mora, Luján (2001) explica que un sistema web es una herramienta capaz de procesar datos a través de peticiones realizadas a un servidor, estas herramientas pueden ser de gran ayuda debido a que son configuradas y creadas de acuerdo a las necesidades del usuario.

Los sistemas web son populares debido a lo práctico del navegador y la independencia del sistema operativo, así como a la facilidad para actualizar y mantenerlos sin distribuir e instalar software a miles de usuarios potenciales. Existen sistemas como los webmails, wikis, weblogs, tiendas en línea y la propia Wikipedia que son ejemplos bastante conocidos.

Es importante mencionar que un Sistema Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que el sistema responderá a cada una de sus acciones, por ejemplo, rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

4.3 Servidores

Tanenbaum y David J (2005) menciona que los servidores son aquellos dispositivos de red que brindan un servicio a otros dispositivos, a los cuales se les conoce como clientes. En general quien realiza esta tarea es un software especializado, pero comúnmente se conoce como servidor al equipo físico donde se ejecuta, el cual es el centro de la infraestructura de la red.

En redes pequeñas es común que un equipo brinde varios servicios simultáneamente, por ejemplo, un servidor de archivos el cual también es servidor de impresión. Partiendo de esta definición, cualquier computadora en la red puede ser un servidor sin necesidad de contar con un hardware o software en particular; aunque existen sistemas operativos especializados (como Microsoft Windows Server, Debian GNU/Linux y SUN Solaris, entre otros) los cuales fueron diseñados específicamente para optimizar los recursos que se comparten a la red. De la misma manera, existen equipos puntualmente creados para funcionar con grandes volúmenes de información, durante las 24 horas y con mejor rendimiento y velocidad que el hardware común de escritorio.

Augusto (2011) agrega que los servidores son equipos informáticos que brindan un servicio en la red. Dan información a otros servidores y a los usuarios. Son equipos de mayores prestaciones y dimensiones que una PC de escritorio. Una computadora común tiene un solo procesador, a veces de varios núcleos, pero uno solo. Incluye un disco rígido para el almacenamiento de datos con una capacidad de 250 GB a 300 GB, en tanto que la memoria RAM suele ser de 2 a 16 GB. Un servidor, en cambio, suele ser mucho más potente. Puede tener varios procesadores con varios núcleos cada uno; incluye grandes cantidades de memoria RAM, entre 16 GB a 1 TB, o más; mientras que el espacio de almacenamiento ya no se limita a un disco duro, sino que puede haber varios de ellos, con capacidad del orden del TB. Debido a sus capacidades, un servidor puede dar un solo servicio o más de uno.

4.3.1 Servidor Web

Tanenbaum y David J (2005) definen que los servidores web son aquellos cuya tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo HTTP (Hypertext Transfer Protocol)

Un servidor WEB consta de un intérprete HTTP el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla.

Además, los servidores pueden disponer de un intérprete de otros lenguajes de programación que ejecutan código embebido dentro del código HTML (Hypertext Markup Language) de las páginas que contiene el sitio antes de enviar el resultado al cliente. Esto se conoce como programación de lado del servidor y utiliza lenguajes como ASP, PHP, Perl y Ajax. La ventaja de utilizar estos lenguajes radica en la potencia de los mismos ejecutando tareas complejas, por ejemplo, acceder a bases de datos abstrayendo al cliente de toda la operación.

4.3.2 Servidor de base de datos

Menéndez, (2011) explica que un servidor de bases de datos brinda servicios de almacenamiento y gestión de bases de datos a sus clientes. Una base de datos es un sistema que permite almacenar grandes cantidades de información. Por ejemplo, todos los datos de los clientes de un banco y sus movimientos en las cuentas. Las bases de datos están situadas en un servidor y se puede acceder a ellas desde terminales o equipos con un programa -llamado cliente- que permita el acceso a la base o bases de datos. Los gestores de base de datos de este tipo permiten que varios usuarios hagan operaciones sobre ella al mismo tiempo.

Los servidores de bases de datos surgen con motivo de la necesidad de las empresas de manejar grandes y complejos volúmenes de datos, al tiempo que requieren compartir la información con un conjunto de clientes (que pueden ser tanto aplicaciones como usuarios) de una manera segura. Ante este enfoque, un sistema gestor de bases de datos (SGBD) deberá ofrecer soluciones de forma fiable, rentable y de alto rendimiento. A estas tres características, se añade una más: debe proporcionar servicios de forma global y, en la medida de lo posible, independiente de la plataforma. Internet se ha convertido en hoy en día en la mayor plataforma de comunicaciones jamás vista. Esto hace que las empresas tiendan a presentar su información a través de la Web en forma de contenidos, que después los clientes consultarán para establecer relaciones con dichas empresas.

Una de las funciones que se empieza a exigir a los SGBD, puesto que sobre ellos recae el peso del almacén y proceso de la información, es la de proporcionar herramientas de apoyo a toma de decisiones ("datawarehouse") al tiempo que proporciona una plataforma de transacciones "on-line" (OLTP) que hacen que la información esté siempre actualizada y consistente.

Aunque parece clara la función de un SGBD, en la actualidad cada vez más filosofías y tecnologías tienden a confluir en un mismo punto. Ya se está hablando acerca de las posibilidades de los nuevos SGBD de poder almacenar contenidos multimedia, objetos o documentos complejos. La explosión de nuevos servicios ha hecho que cada vez más aplicaciones dependan de estos servidores de datos, delegando la responsabilidad de la gestión y almacenamiento de la información a aquellos que mejor están preparados para su tratamiento.

4.3.3 Microsoft SQL Server

Silberschatz & Korth (2002) explican que un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a ellos. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de manera que sea tanto práctica como eficiente y gestionar grandes cantidades de la misma.

La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la misma. Además, los SGBD deben proporcionar la fiabilidad de lo que se almacena, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos.

Dado que la información es tan importante en la mayoría de las organizaciones, los científicos informáticos han desarrollado un amplio conjunto de conceptos y técnicas para la gestión de los datos.

Guthrie (2015) comenta que Microsoft SQL server es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado por línea de comandos o mediante la interfaz gráfica de management studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL). Puede ser configurado para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor, y los siguientes nombres específicos (con un guión invertido entre el nombre del servidor y el nombre de la instalación).

Transact-SQL es un lenguaje que utiliza este sistema, es el principal medio de interacción con el Servidor, el cual permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de base de datos, inserción y modificación de datos, así como la administración del servidor. Esto se realiza mediante el envío de sentencias en T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente.

4.4 Protocolo de Transferencia de Hipertexto o HTTP

Wall (2005) explica que Hypertext Transfer Protocol o HTTP (en español Protocolo de Transferencia de Hipertexto) es el protocolo usado en cada transacción de la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, que son una serie de publicaciones que hacen el grupo de trabajo de internet, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.

El protocolo de transferencia de hipertexto está orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). El resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

Una definición más sencilla, la describe Puigvert (2004) que plantea que HTTP es un protocolo utilizado para transferir información en el World Wide Web. Una dirección HTTP (un tipo de localizador de recursos universal o dirección URL), tiene el siguiente formato :

<http://www.lexus.com>

4.5 Internet

Axis (2005) define que Internet es un conjunto descentralizado de redes, es decir, cada conjunto de red está en una ubicación geográfica distinta una de otros, estas redes de comunicación están interconectadas utilizando reglas de comunicación llamadas protocolos; esto permite que se conecten computadoras y usuarios en todo el mundo. Su forma de operación es descentralizada, lo cual significa que la información no necesita pasar necesariamente por un nodo de la red, sino que puede tomar caminos alternativos según convenga.

Sus orígenes se remontan a la década de 1960, dentro de ARPA (hoy DARPA, las siglas en inglés de la Defense Advanced Research Projects Agency), como respuesta a la necesidad de esta organización de buscar mejores maneras de usar los computadores de ese entonces, pero enfrentados al problema de que los principales investigadores y laboratorios deseaban tener sus propios computadores, lo que no solo era más costoso, sino que provocaba una duplicación de esfuerzos y recursos. El verdadero origen de Internet, Así nace ARPANet (Advanced Research Projects Agency Network o Red de la Agencia para los Proyectos de Investigación Avanzada de los Estados Unidos), que legó el trazado de una red inicial de comunicaciones de alta velocidad a la cual fueron integrándose otras instituciones gubernamentales y redes académicas durante los años 70, 80 y 90s.

Puigvert (2004) comenta, Internet es un “Mundo” de información, al que se puede acceder en cualquier parte del planeta, mediante una computadora y una conexión, sea por fibra óptica, cable coaxial o línea telefónica.

4.6 Base de Datos

Valdés (2007) comenta que una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, y por ende, se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento.

Sudarshan (2011) explica que una base de datos (cuya abreviatura es BD) es una entidad en la cual se puede almacenar información de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizarlos. Por lo tanto, el concepto generalmente está relacionado con el de red ya que se debe poder compartir esta información. Una BD proporciona a los usuarios el acceso a lo que se guarda para que se pueda visualizar, ingresar o actualizar, en concordancia con los derechos de acceso que se les hayan otorgado. Se convierte más útil en la medida que la cantidad de datos almacenados crece.

Una base de datos puede ser local, es decir que puede utilizarla sólo un usuario en un equipo, o puede ser distribuida, es decir que la información se almacena en equipos remotos y se puede acceder a ella a través de una red. La principal ventaja de utilizar bases de datos es que múltiples usuarios pueden acceder a ellas al mismo tiempo.

Una consulta es el método para acceder a los datos almacenados. Las consultas se realizan a través de un Lenguaje de manipulación de datos, a través de estas, se puede modificar, borrar, mostrar y agregar datos, también pueden utilizarse como origen de registro para formularios.

4.7 Framework

Según Sanchez (2006), un Framework es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. Sí, es una definición muy genérica, pero también puede serlo un framework: sin ir más lejos, el paradigma MVC (Model-View-Controller) dice poco más que “separa en tu aplicación la gestión de los datos, las operaciones, y la presentación”. En el otro extremo, otros frameworks pueden llegar al detalle de definir los nombres de ficheros, su estructura, las convenciones de programación, etc.

Los frameworks tienen como objetivo principal ofrecer una funcionalidad definida, auto contenido, siendo contruidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento. Para acceder a esa funcionalidad, se construyen piezas, objetos, llamados objetos calientes, que vinculan las necesidades del sistema con la funcionalidad que este presta.

Esta funcionalidad, está constituida por objetos llamados fríos, que sufren poco o ningún cambio en la vida del framework, permitiendo la portabilidad entre distintos sistemas. Lo esencial para ser denominados frameworks es estar constituidos por objetos casi estáticos con funcionalidad definida a nivel grupo de objetos y no como parte constitutiva de estos, por ejemplo, en sus métodos, en cuyo caso se habla de un API o librería.

Casi todos los frameworks comparten las mismas características de acuerdo a su tipo, entre ellas se pueden destacar:

- La Autenticación mediante login y password que permite restringir el acceso y el tipo de permiso.
- El Acceso a los datos en archivos txt, xml, mediante interfaces que integran la base de datos.
- Abstracción de URLs y Sesiones ya que el framework se encarga de manejarlas.
- Internacionalización que permite la inclusión de varios idiomas en el desarrollo.
- Controladores fácilmente adaptables a las necesidades del proyecto que gestionan las peticiones y/o eventos.

El uso de frameworks permite a los desarrolladores ser más productivos, es decir ejecutar más proyectos, generar más ingresos y menos tiempo de trabajo.

4.8 ASP.NET

Según Guthrie (2007) ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Microsoft introdujo la tecnología llamada Active Server Pages en diciembre de 1996. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente. La definición contextual de Microsoft es que "Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web".

Después del lanzamiento del Internet Information Services 4.0 en 1997, Microsoft comenzó a investigar las posibilidades para un nuevo modelo de aplicaciones web que pudiera resolver las quejas comunes sobre ASP, especialmente aquellas con respecto a la separación de la presentación y el contenido y ser capaz de escribir código "limpio". A Mark Anders, un administrador del equipo de IIS y Scott Guthrie, quien se había unido a Microsoft en 1997 después de graduarse de la Universidad Duke, se les dio la tarea de determinar cómo debería ser ese modelo. El diseño inicial fue desarrollado en el transcurso de dos meses por Anders y Guthrie, Guthrie codificó los prototipos iniciales durante las celebraciones navideñas de 1997.

La primera demostración pública y la liberación de la primera beta de ASP+ (y el resto del .NET Framework) se realizó en el Microsoft's Professional Developers Conference (PDC) el 11 de julio de 2000 en Orlando, Florida. Durante la presentación de Bill Gates, Fujitsu demostró ASP+ usado en conjunción con COBOL 5 y el soporte para una variedad de otros lenguajes fue anunciada, incluyendo los nuevos lenguajes de Microsoft, Visual Basic .NET y C#, así como también el soporte por medio de herramientas de interoperabilidad para Python y Perl creadas por la empresa canadiense ActiveState.

4.9 Lenguaje Unificado de Modelado o UML

Según Rob Pooley (2002), UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Este lenguaje se utiliza cuando se describe un proceso o método en el sistema, el usuario tiene una mejor perspectiva de cómo funciona un determinado proceso. Se logra apreciar cómo se invocan mensajes, cómo se comunican funciones con el usuario, y se detalla paso a paso los procesos.

Según booch (2004) , UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y una reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Este lenguaje indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo. Las objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

Diagramas UML

(Kendall & E, 2005) Habla sobre los principales diagramas UML:

Diagrama de caso de uso

Un diagrama de caso de uso narra una historia estilizada sobre cómo interactúa un usuario final (Que tiene cierto número de roles posibles) con el sistema. El primer paso para escribir un caso de uso es definir un conjunto de “actores” que estarán involucrados en la historia. Los actores son las distintas personas (o dispositivos) que usan el sistema o producto en el contexto de la función y comportamiento que va a describirse. Los actores representan los papeles que desempeñan las personas (o dispositivos) cuando opera el sistema. Con una definición más formal, un actor es cualquier cosa que se comunique con el sistema o producto y que sea externo a éste.

Diagrama de actividades

Los diagramas de actividades muestran las secuencias de actividades de un proceso, incluyendo las actividades secuenciales, las actividades paralelas y las decisiones que se toman. Por lo general, un diagrama de actividades se elabora para un caso de uso y podría reflejar los diferentes escenarios posibles.

Diagrama de secuencia

Los diagramas de secuencias pueden ilustrar una sucesión de interacciones entre clases o instancias de objetos en un período determinado. Los diagramas de secuencias se utilizan con frecuencia para representar el proceso descrito en los escenarios de caso de uso.

Diagrama de contexto

El diagrama de contexto es un caso especial del diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema. El diagrama de contexto muestra a través de flujos de datos las interacciones existentes entre los agentes externos y el sistema, sin describir en ningún momento la estructura del sistema

de información. El diagrama de contexto consiste de terminadores, flujos de datos y flujos de control, almacenes de datos y un solo proceso, que consiste en una sola burbuja. El nombre dentro del proceso suele ser el nombre del sistema completo o un acrónimo convenido.

Los terminadores se representan por medio de rectángulos y se comunican con el sistema utilizando flujos de datos o de control, los cuales son representados por flechas, o a través de almacenes externos. Hay que recalcar que los terminadores no se comunican entre sí, al menos no en el diagrama de contexto, ya que denotarían interacciones externas al sistema.

4.10 Ingeniería de Software

Todos los conceptos y herramientas que se han mencionado se rigen bajo la ingeniería del software. La ingeniería del software es una aplicación de estándares y teorías al desarrollo de un proyecto informático, tiene que ver con la seguridad, la calidad del producto y cómo es la organización del equipo que desarrolla el software.

Según Roger S. Pressman (2010) la ingeniería de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto. La ingeniería de software, por lo tanto, incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema. Cabe destacar que, el proceso de desarrollo de software implica lo que se conoce como ciclo de vida del software, que está formado por cuatro etapas: concepción, elaboración, construcción y transición.

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) en 1993, ha desarrollado una definición más completa.

La ingeniería de software es:

- La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software. Es decir se debe seguir una serie de pasos donde de manera ordenada, se vaya estructurando el producto final, respetando cada fase. Esto ayuda a seguir un plan donde se hace un diseño antes de empezar a codificar, esto evita errores de implementación y un mayor entendimiento de las necesidades del cliente.

4.11 Proceso del software

Roger S. Pressman (2010) en su obra Ingeniería del software expone que cuando se inicia la construcción de un sistema o producto, es importante ejecutar una serie de pasos predecibles, como un mapa a seguir en el camino de la elaboración del producto final deseado. A este mapa o pasos predecibles se le llama proceso de software.

Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. Una actividad busca lograr un objetivo amplio (por ejemplo, comunicación con los participantes) y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software. Una acción (diseño de la arquitectura) es un conjunto de tareas que producen un producto importante del trabajo (por ejemplo, un modelo del diseño de la arquitectura). Una tarea se centra en un objetivo pequeño, pero bien definido (por ejemplo, realizar una prueba unitaria) que produce un resultado tangible.

En el contexto de la ingeniería de software, un proceso no es una prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán.

La estructura del proceso establece el fundamento para el proceso completo de la ingeniería de software por medio de la identificación de un número pequeño de actividades estructurales que sean aplicables a todos los proyectos de software, sin importar su tamaño o complejidad. Además, la estructura del proceso incluye un conjunto de actividades sombrilla, que son aplicables a través de todo el proceso del software, estas actividades son para garantizar el correcto funcionamiento del software y mitigar lo más posible los errores.

Una estructura de proceso general para la ingeniería de software consta de cinco actividades:

- **Comunicación:** Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software.
- **Planeación:** Cualquier viaje complicado se simplifica si existe un mapa. Un proyecto de software es un viaje difícil, y la actividad de planeación crea un “mapa” que guía al equipo mientras viaja. El mapa —llamado plan del proyecto de software— define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos

que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.

- **Modelado:** independientemente de la profesión que se tenga, por lo general, a diario trabaja con modelos. Se crea un “bosquejo” del objeto por hacer a fin de entender el panorama general —cómo se verá arquitectónicamente, cómo ajustan entre sí las partes constituyentes y muchas características más—. Si se requiere, se refina el bosquejo con más y más detalles en un esfuerzo por comprender mejor el problema y cómo resolverlo. Un ingeniero de software hace lo mismo al crear modelos a fin de entender mejor los requerimientos del software y el diseño que cubran las expectativas del cliente.
- **Construcción:** Esta actividad combina la generación de código (ya sea manual o automatizado) y las pruebas que se requieren para descubrir errores en éste.
- **Despliegue:** El software (como entidad completa o bien un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y le da retroalimentación, misma que se basa en dicha evaluación.

Estas cinco actividades estructurales genéricas se usan durante el desarrollo de programas pequeños y sencillos, en la creación de aplicaciones web grandes y en la ingeniería de sistemas enormes y complejos basados en computadoras. Los detalles del proceso de software serán distintos en cada caso, pero las actividades estructurales son las mismas.

4.12 Metodología de desarrollo ágil SCRUM

Roger S. Pressman (2010) expresa, SCRUM (nombre que proviene de cierta jugada que tiene lugar durante un partido de rugby) es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990.

Los principios SCRUM son congruentes con el manifiesto ágil, este manifiesto que fue acuñado en 2001 en una reunión por 17 críticos expertos en software son principios resumidos de metodologías alternativas a las existentes que pretenden agilizar el proceso de desarrollo, y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega.

Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint. El trabajo realizado dentro de un sprint (el número de éstos que requiere cada actividad estructural variará en función de la complejidad y tamaño del producto) se adapta al problema en cuestión y se define —y con frecuencia se modifica— en tiempo real por parte del equipo SCRUM.

SCRUM acentúa el uso de un conjunto de patrones de proceso del software que han demostrado ser eficaces para proyectos con plazos de entrega muy apretados, requerimientos cambiantes y negocios críticos. Cada uno de estos patrones de proceso define un grupo de acciones de desarrollo:

Product Backlog: es el conjunto de requisitos funcionales y no funcionales que debe cumplir el producto una vez entregado. No se requiere que esté completo al momento de su creación, basta con definir aquellos requisitos que se conozcan en su momento y alentar a su crecimiento continuo o su modificación.

Sprints: consiste en unidades de trabajo que se necesitan para alcanzar un requerimiento definido que debe ajustarse en una caja de tiempo predefinida (lo común son 30 días, pero el tiempo siempre lo puede establecer el equipo de trabajo). Durante el sprint no se introducen cambios (por ejemplo, aspectos del

trabajo retrasado). Así, el sprint permite a los miembros del equipo trabajar en un ambiente de corto plazo pero estable.

Reuniones SCRUM: son reuniones breves (de 15 minutos, por lo general) que el equipo SCRUM efectúa a diario. Hay tres preguntas clave que se pide que respondan todos los miembros del equipo:

- ¿Qué hiciste desde la última reunión del equipo?
- ¿Qué obstáculos estás encontrando?
- ¿Qué planeas hacer mientras llega la siguiente reunión del equipo?

Un líder del equipo, llamado maestro SCRUM, dirige la junta y evalúa las respuestas de cada persona. La junta SCRUM ayuda al equipo a descubrir los problemas potenciales tan pronto como sea posible. Así mismo, estas juntas diarias llevan a la “socialización del conocimiento”, con lo que se promueve una estructura de equipo con organización propia. Demostraciones preliminares: entregar el incremento de software al cliente de modo que la funcionalidad que se haya implementado pueda demostrarse al cliente y éste pueda evaluarla.

Historias de usuario: Una historia de usuario es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario.

En el siguiente diagrama se resume la metodología de SCRUM, con los

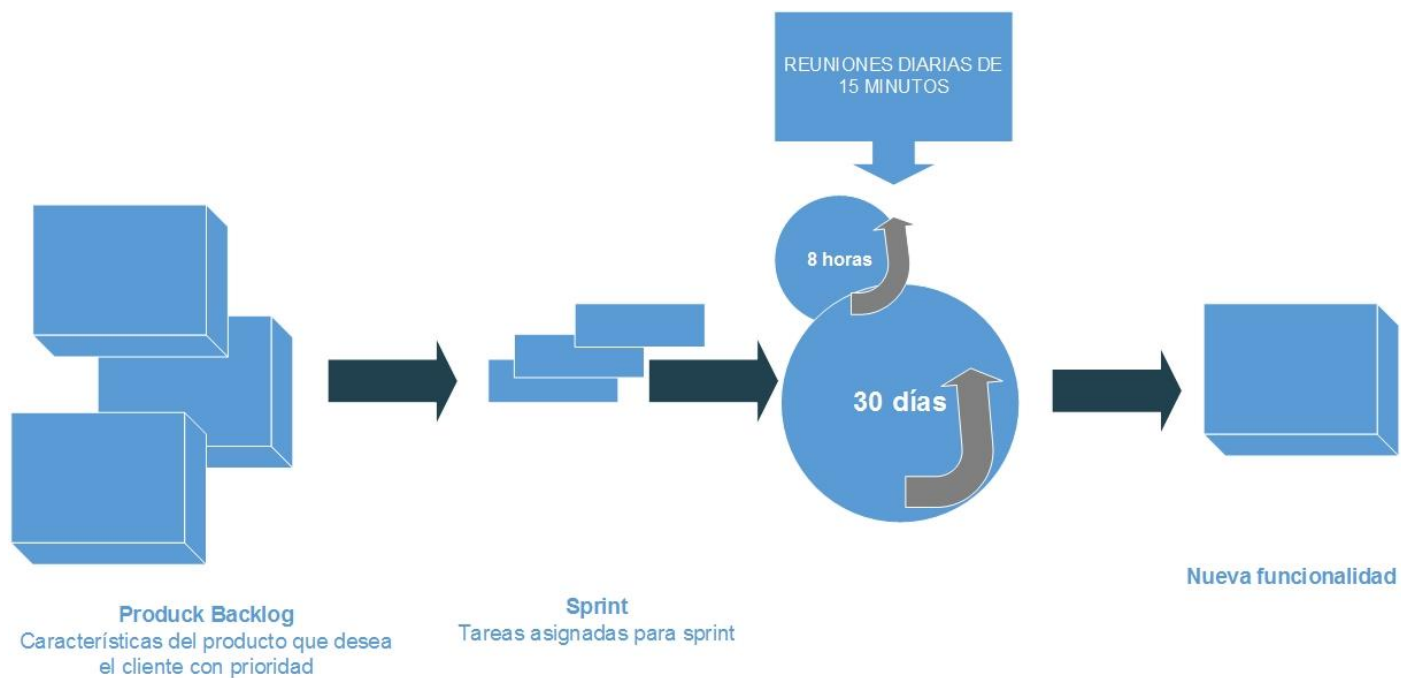


Figura 2: Diagrama proceso SCRUM (Elaboración propia)

5. Desarrollo de la solución propuesta

5.1 Tipo de Investigación

Acorde a lo planteado por Pressman (2010), la investigación desarrollada fue eminentemente aplicada, particularmente porque requirió la aplicación de ciertos conocimientos de la Ingeniería en Computación en el desarrollo de un sistema de información para la Vivienda Estudiantil, en la Universidad Nacional de Ingeniería ubicada en la ciudad de Managua.

Se hizo uso de los conocimientos y habilidades obtenidas a lo largo de la carrera de Ingeniería en Computación, como metodología de desarrollo de software, que permitió hacer el análisis, diseño y desarrollo de los procesos antes descritos de la Vivienda Estudiantil. Otro aspecto significativo fue el empleo de teorías de elaboración de estudio de factibilidad, en la realización de los estudios de factibilidad, económica, técnica y operativa. Así mismo, se utilizaron referencias de la administración de base de datos, normalización de tablas y nomenclatura de nombres para la elaboración y diseño de la base de datos, paradigma orientado a objetos, estructuras de datos, estructuras condicionales, ciclos, y algoritmización para la codificación del sistema de información web.

5.2 Diseño de Instrumentos de medición

Para obtener información se hizo uso de instrumentos de medición, como la observación y la entrevista, posteriormente se analizó la información para sacar las conclusiones.

A continuación se detallan los parámetros que se tomaron en cuenta al momento de elaborar las preguntas de la entrevista y el proceso de observación.

Entrevista

Según Pressman (2010), la entrevista es el enfoque más directo de conocer los requerimientos. Los miembros del equipo de software se reúnen con los usuarios para entender mejor sus necesidades, motivaciones, cultura laboral y una multitud de aspectos adicionales. Esto se logra en reuniones individuales o a través de grupos de enfoque.

Una entrevista es un intercambio de ideas, opiniones mediante una conversación que se da entre una, dos o más personas donde un entrevistador es el designado para preguntar

Parámetros para elaborar entrevista

- Se deben hacer preguntas del tipo abierta, para permitir a las personas entrevistadas indiquen su punto de vista con respecto a algún aspecto en especial.
- Las preguntas deben ser claras y comprensibles para el entrevistado.
- Elaborar las preguntas de acuerdo al lenguaje característico del entrevistado.
- Evaluar si el entrevistado puede y quiere aportar la información que se le solicita.
- Se permite hacer respuestas cerradas, pero se deben evitar el uso excesivo de las mismas.
- Evaluar la pertinencia de la pregunta.

Ver entrevista elaborada (**Anexo 8: Entrevista de levantado de requerimientos**)

Observación

Kendall (2005), dice que observación es la acción y efecto de observar (examinar con atención, mirar con recato, advertir). Se trata de una actividad realizada por los seres vivos para detectar y asimilar información. La misma nos permite comprobar un determinado proceso, sin interferir con las tareas realizadas por el cliente o usuario.

La técnica de observación permitió conocer, de forma directa, el objeto de estudio para luego describir y analizar los procesos de administración de la Vivienda Estudiantil de una manera directa en el entorno de estudio.

Elementos que conformaron el proceso de observación

Los sujetos y objetos que constituyen el proceso de observación deben de ser claramente definidos, en el caso de esta investigación se determinaron los siguientes:

- El sujeto que se investiga (encargado de la administración de la Vivienda Estudiantil)
- El objeto de estudio (los procesos administrativos de la Vivienda Estudiantil)
- Los medios en los que se da la observación (Vivienda Estudiantil)
- Los instrumentos a utilizar (computadora portátil, utilizada para documentar los pasos realizados en cada proceso administrativo).

Descripción Técnica	Instrumentos	Resultados esperados
Entrevista	Cuestionario elaboración propia, herramienta de google, notas.	- Requerimientos para el sistema de información.
Observación	Guía de procedimiento, Lista de chequeo.	- Entender mejor cómo se llevan a cabo algunos procesos administrativos de la Vivienda Estudiantil.

Tabla 1: Descripción de elementos de medición (Elaboración propia)

5.3 Delimitación geográfica

El proyecto de sistema web para la gestión de la Vivienda Estudiantil y los becados residentes se realizó en la ciudad de Managua, municipio de Managua. En la Universidad Nacional de Ingeniería, en el Recinto Universitario Simón Bolívar, donde se encuentra ubicado el edificio de la Vivienda Estudiantil.



Figura 3: Ubicación geográfica Universidad Nacional De Ingeniería (fuente Google Maps)

En el siguiente acápite se describe el proceso que se desarrolló paso a paso, durante todo el proyecto.

6. Desarrollo del Sistema de información web

Como parte inicial del desarrollo del sistema de información web para la Vivienda Estudiantil, se realizó un estudio de factibilidad para comprobar la viabilidad del proyecto. Posteriormente se planificaron reuniones con el encargado de la Vivienda Estudiantil, para llevar a cabo la entrevista y conocer los requerimientos y el proceso de observación. Consecutivamente se construyó el diseño y la lógica. Una vez obtenido el diseño se codificó el sistema de información web, y se efectuaron reuniones de presentación de resultados con el encargado.

A continuación, se detalla cada paso de manera específica:

Se muestran los resultados del estudio de factibilidad donde se analizaron distintos aspectos a fin de valorar el éxito del proyecto y decidir su realización.

6.1 Estudio de factibilidad

Como parte de la investigación que se realizó se elaboró un estudio de factibilidad, donde se pudo estudiar la viabilidad del proyecto, el cual consistió en el desarrollo de un sistema de información web para la Vivienda Estudiantil.

En este estudio se logró evaluar la factibilidad técnica, operativa, de cronograma, legal y económica. (**Ver Anexo 2: Estudio de factibilidad**)

Como primeros puntos se describió la situación actual de la Vivienda Estudiantil, se explicó cómo se realizan los procesos manuales, se plantearon los beneficios del proyecto y las restricciones del mismo.

A continuación, se describe las conclusiones de cada estudio de factibilidad elaborado.

- **Factibilidad Técnica:**

En la factibilidad técnica se evaluaron dos aspectos relevantes, requeridos para el desarrollo del proyecto: el hardware y software.

Como punto importante en la investigación previa a la implementación del proyecto de desarrollo del sistema de información web, se indagó si se contaba con los equipos adecuados y el personal para llevar a cabo el proyecto.

Los resultados de la factibilidad técnica fueron positivos, ya que la Vivienda Estudiantil de los becados cuenta con el apoyo de la Universidad Nacional de Ingeniería (UNI). La UNI como institución tecnológica cuenta con equipos existentes de desarrollo de software o proyectos afines al de la Vivienda Estudiantil.

El equipo de desarrollo levantó un detalle de los equipos disponibles y se concluyó que tienen las características necesarias para la implementación del proyecto de desarrollo del sistema de información web para la Vivienda Estudiantil. (**Ver anexo 3 Factibilidad Técnica**).

- **Factibilidad Operativa:**

En la factibilidad operativa se analizó si el proyecto se puede poner en marcha y si se cuenta con personal capacitado que pueda aprender el uso del sistema de información web para la Vivienda Estudiantil. Como se mencionó anteriormente, una de las grandes ventajas que existe en este proyecto es que cuenta con el apoyo de la UNI. Dicha institución ha puesto a la orden un personal profesional en las áreas informáticas, capaces de operar el sistema de información web una vez finalizado. **(Ver Anexo 4: factibilidad operativa)**

- **Factibilidad de cronograma**

En la factibilidad de cronograma se elaboró un plan que contempló los tiempos razonables para la implementación del proyecto de desarrollo del sistema de información web. Se utilizó la metodología COCOMO II¹, como un enfoque para determinar la duración del proyecto en tiempo. Se estimó un tiempo razonable de 8 meses. **(Ver Anexo 5: Factibilidad de Cronograma)**

- **Factibilidad legal**

En la factibilidad legal se evaluó si en la implementación del proyecto de propuesta de desarrollo del sistema de información web se podría infringir alguna ley. Como puntos destacables, cabe mencionar que, en cuanto a licencias de software, se cuenta con legalidad de uso, gracias a la compra de licencias realizadas por la Universidad Nacional de Ingeniería.

¹ En su libro clásico acerca de “economía de la ingeniería de software”, Barry Boehm [Boe81] introdujo una jerarquía de modelos de estimación de software que llevan el nombre COCOMO, por COConstructive COst MOdel: modelo constructivo de costos. El modelo COCOMO original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria. Evolucionó hacia un modelo de estimación más exhaustivo, llamado COCOMO II [Boe00]. **(Roger S. Pressman, 2010)**

Se implementó control de accesos para proteger la información personal de los estudiantes y administradores de la Vivienda Estudiantil que se encuentran en la base de datos del sistema propuesto, esto para evitar que, personas no autorizadas tengan acceso a los datos sensibles de los usuarios. (**Ver Anexo 6: Factibilidad Legal**)

- **Factibilidad económica**

En este acápite, se analizó el impacto financiero que tiene el proyecto de desarrollo del sistema de información web para la Vivienda Estudiantil, del cual se concluyó que, el proyecto es viable, debido a que, se cuenta con los recursos económicos, humanos y tecnológicos necesarios para su correcta implementación, además, se detallan los ahorros económicos derivados del desarrollo del proyecto. (**Ver Anexo 7: Factibilidad Económica**)

6.2 Metodología de Desarrollo SCRUM

El modelo de desarrollo que se aplicó en el sistema de información web para la gestión de la Vivienda Estudiantil y becados residentes de la UNI, fue la Metodología Ágil SCRUM.

Como se explicó en el marco teórico, en SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al propietario del proyecto. Por ello, SCRUM está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados a corto plazo, donde

los requisitos son cambiantes y la innovación, la competitividad, la flexibilidad y productividad son fundamentales.

A continuación, se presenta cada una de las etapas del proyecto:

6.2.1 Sprint backlog y sprint planning meeting²

En esta fase, se aplicó el estándar ISO 9000:2000 para aseguramiento de la calidad del software (**Ver anexo 10: Lista de ISO utilizadas**), donde se efectuaron estimaciones de tiempos de entrega y revisiones de la funcionalidad del software, se determinó el equipo de desarrollo, los backlogs y las tareas a desarrollar en cada sprint.

En la siguiente tabla se detalla el personal involucrado en el proyecto del desarrollo del sistema de información web.

Nombre	Contacto	Rol
Encargado de la Vivienda Estudiantil	89764918	Product Owner/Cliente
Francel Martínez	88577365	Scrum Team
Kenneth Gaitán	77790299	Scrum Team

Tabla 2: Equipo de Desarrollo del sistema de información web

² Lista de tareas y planificación de la iteración

Se utilizó la herramienta Microsoft Project, para elaborar cada Sprint y se acordó la duración que tendría cada uno de estos en todo el proceso de desarrollo. Durante cada sprint, se abordaron diferentes características del sistema y la prioridad de finalización que tenía cada una de estas, además del diseño de bases de datos, diseño de interfaz y algoritmización de los principales procesos que se automatizaron.

El desarrollo del Sistema web de Información para la Vivienda Estudiantil de la UNI, se llevó a cabo en catorce sprints, con una duración de dos semanas cada uno.

A continuación, se muestra el detalle de los sprints y los backlogs, ejecutados a lo largo del proceso de desarrollo, conforme a los lineamientos de la Metodología Ágil SCRUM y los estándares de desarrollo de software aplicados.

	Task Name	Duration	Start	Finish	Predecessors
	Agile project management	140 days	Mon 21/3/16	Fri 30/9/16	
✓	Sprint #1 - Identificación de Necesidades y Requerimientos	2 wks	Mon 21/3/16	Fri 1/4/16	
✓	Sprint #2 - Requerimientos Funcionales y Diseños Preliminares	2 wks	Mon 4/4/16	Fri 15/4/16	1
✓	Sprint #3 - Diseño de Base de Datos	2 wks	Mon 18/4/16	Fri 29/4/16	2
✓	Sprint #4 - Mantenimiento de Catálogos	2 wks	Mon 2/5/16	Fri 13/5/16	3
✓	Sprint #5 - Módulo de Estudiantes	2 wks	Mon 16/5/16	Fri 27/5/16	4
✓	Sprint #6 - Módulo de Alimentación - Configuración de Horarios y costos	2 wks	Mon 30/5/16	Fri 10/6/16	5
✓	Sprint #7 - Módulo de Alimentación - Validación de Entrega de Alimentos	2 wks	Mon 13/6/16	Fri 24/6/16	6
✓	Sprint #8 - Módulo de Limpieza	2 wks	Mon 27/6/16	Fri 8/7/16	7
✓	Sprint #9 - Módulo de Inventario - Mantenimiento de Artículos	2 wks	Mon 11/7/16	Fri 22/7/16	8
✓	Sprint #10 - Módulo de Inventario - Movimientos de Inventario	2 wks	Mon 25/7/16	Fri 5/8/16	9
✓	Sprint #11 - Módulo de Actividades y Servicio Social	2 wks	Mon 8/8/16	Fri 19/8/16	10
✓	Sprint #12 - Módulo de Memorándums	2 wks	Mon 22/8/16	Fri 2/9/16	11
✓	Sprint #13 - Pruebas	2 wks	Mon 5/9/16	Fri 16/9/16	12
✓	Sprint #14 - Capacitación e Implementación	2 wks	Mon 19/9/16	Fri 30/9/16	13

Figura 4: Listado de Sprints

6.2.1.1 Sprint #1 - Identificación de las Necesidades y Requerimientos del Sistema

En este primer sprint, se elaboró y se construyó un wish list (lista de deseo), sobre las funcionalidades requeridas del sistema, en conjunto con el Product Owner (el dueño del proyecto), encargado de la Vivienda Estudiantil de la UNI.

Se realizaron reuniones con el usuario, para identificar cómo funcionaban los procesos existentes y los problemas que había en la Vivienda Estudiantil de la UNI. El Product Owner respondió a la pregunta ¿Qué hay que hacer o qué debe hacer el sistema?

Se aplicó una entrevista (**ver anexo 8: Entrevista de levantado de requerimiento**) como instrumento de recolección de información, para ayudar en este proceso de levantamiento de requerimientos.

6.2.1.2 Sprint #2 - Definición de Requerimientos del Sistema

Una vez recabada la información, se analizaron datos obtenidos y se definieron los requerimientos funcionales a través la elaboración de historias de usuario, cabe destacar que cada proceso también se validó con dos sesiones de observación presencial (**Ver Anexo 9: Documento de proceso de verificación de observación**).

Las historias de usuario son descripciones cortas y simples de una funcionalidad, escritas desde la perspectiva de la persona que necesita una nueva capacidad de un sistema, por lo general el usuario, área de negocio o cliente. A continuación se presentan las historias de usuarios.

6.2.1.2.1 Historias de Usuario

Como resultado de las entrevistas de levantamiento de requerimientos, se formularon las siguientes historias de usuarios:

Módulo de Alimentación:

HU-MA-01 – Configurar Horarios de Alimentación

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Configurar Horarios de Alimentación, en los cuales los estudiantes becados residentes retiren sus alimentos en los diferentes comedores de la UNI

Para: Permitir la validación de entrega de servicios alimenticios en los horarios establecidos.

HU-MA-02 – Configurar Costos de Servicios Alimenticios

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Configurar los costos de los servicios alimenticios (por ejemplo, costo de desayuno, almuerzo y cena)

Para: Manejar los costos de inversión en conceptos de alimentación que se realizan cada semestre en la Vivienda Estudiantil.

HU-MA-03 – Revisar listado de entrega de servicios alimenticios

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Ver el listado de alimentos entregados diariamente a cada estudiante becado residente

Para: Poder generar reportes de entrega de alimentos.

HU-MA-04 – Validar entrega de servicios alimenticios a los estudiantes residentes

Cómo: Despachador de Comedor y Usuario autorizado en el sistema

Quiero: Ingresar número de carnet de estudiante y verificar si éste tiene autorización para retirar el servicio alimenticio en los diferentes horarios establecidos por la Vivienda Estudiantil de la UNI

Para: Poder entregar el servicio alimenticio al estudiante autorizado que la solicita.

HU-MA-05 – Autorizar manualmente el retiro de alimentos a Estudiantes

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Autorizar manualmente el retiro de alimentos a estudiantes que, por motivos de fuerza mayor, no hayan podido hacerlo en tiempo y forma.

Para: Llevar el control ordenado de los servicios alimenticios que se retiran fuera de tiempo y evitar inconsistencias en los reportes.

HU-MA-06 – Registro de Recintos universitarios y comedores disponibles

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Registrar los recintos universitarios y los comedores que están disponibles en el sistema

Para: para poder configurar los módulos del sistema que requieren recintos

Módulo de inventario:

HU-MI-01 – Registro de enseres / inventario nuevo en bodega

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Ingresar artículos y las cantidades correspondientes de los mismos en el inventario.

Para: mantener registro de las altas/entradas de activos fijos que se realizan en la Vivienda Estudiantil.

HU-MI-02 – Asignación de artículos de inventario por estudiante y por cuarto

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Seleccionar artículos del inventario y asignarlos a un estudiante o cuarto en específico.

Para: generar movimientos de inventario en concepto de asignación, al estudiante o al cuarto de la Vivienda Estudiantil que lo necesite.

HU-MI-03 – Baja de artículos por cuarto y estudiantes

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Dar de baja a los artículos en inventario que han agotado su existencia, han sido entregados o asignados a estudiantes y/o cuartos

Para: mantener el soporte de los movimientos de activos fijos de la Vivienda Estudiantil de la UNI.

Módulo de Actividades:

HU-MSS-01 – Crear actividades de servicio social

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Quiero crear una agenda de actividades sociales e invitar, por medio de correo electrónico, a los estudiantes.

Para: Llevar el control de asistencia y cumplimiento de los estudiantes con respecto a las labores sociales organizadas por la Vivienda Estudiantil.

HU1-MSS-02 – Asignación de actividades por estudiantes

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Asignar número definido de estudiantes a una actividad social creada.

Para: Llevar registros de la participación de cada estudiante.

HU-MSS-03 – Ingreso de horas de trabajo voluntario por estudiante

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Ingresar la cantidad de tiempo que un estudiante ha dedicado a una actividad social a la cual ha sido invitado.

Para: Llevar control de cumplimiento de actividades por cada estudiante.

Módulo de Encuestas:

HU-ME-01 – Creación y activación de encuestas

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Crear una encuesta, publicarla y enviar invitación por correo a los estudiantes seleccionados, para que éstos puedan contestar.

Para: Recolectar las diferentes opiniones de los estudiantes, por ejemplo, ¿qué piensan acerca de la calidad de la atención recibida en los comedores de la UNI?

HU-ME-02 – Contestar encuestas

Cómo: Estudiante Becado Residente, Hospedado en la Vivienda Estudiantil

Quiero: Acceder a las encuestas creadas por la administración de la Vivienda Estudiantil y responder a la misma.

Para: Bridar mi opinión acerca del tema en cuestión.

HU-ME-03 – Ver resultados de las encuestas

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Ver los resultados de las encuestas completadas por los estudiantes.

Para: Conocer la opinión de los estudiantes y utilizar esta información para mejorar algunos procesos de la Vivienda Estudiantil u otros temas abordados en las encuestas.

Módulo de Limpieza

HU-ML-01 – Generar Rol de Aseo por Cuarto

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Generar el rol de aseo por cuarto, para cada semestre

Para: Organizar las tareas asignadas a cada estudiante becado residente y llevar el control del cumplimiento de roles de limpieza.

HU-ML-02– Generar Rol de Aseo por Área

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Generar el rol de aseo de las áreas de la vivienda que le corresponde a limpiar a cada estudiante durante el semestre.

Para: Poder organizar los turnos de limpieza de áreas a cada estudiante.

Módulo de estudiantes:

HU-ME-01 – Asignar cuarto a estudiante

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Asignar a cada estudiante becado residente, el cuarto en el que le corresponde hospedarse en cada semestre.

Para: Distribuir a los estudiantes en los diferentes cuartos de la Vivienda Estudiantil, organizar los roles de limpieza, permitir la asignación de inventario y enseres por cuarto y estudiante.

HU-ME-02 – Enviar Memorando

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Redactar memorandos a los estudiantes que incumplan con las normas establecidas en los reglamentos de comportamiento de la Vivienda Estudiantil y la UNI, enviar dicho memorándum por correo electrónico a los respectivos destinatarios (por ejemplo, estudiante involucrado y autoridades de la Dirección de Bienestar Estudiantil.)

Para: Llevar el control del expediente, comportamiento y reincidencias del estudiante.

HU-ME-03 – Registrar Ayuda Económica Entregada a los estudiantes

Cómo: Administrador de la Vivienda Estudiantil

Quiero: Registrar las cantidades de dinero entregadas a los estudiantes y el concepto de estas transacciones. Además, generar un recibo de entrega.

Para: Llevar el control de estipendios y ayuda económica entregada cada semestre a los estudiantes becados residentes de la Vivienda Estudiantil.

6.2.1.2.2 Requisitos no funcionales

En Ingeniería de Software, los requisitos no funcionales son atributos o propiedades de calidad que el producto debe tener. Especifican criterios que pueden utilizarse para juzgar la operación de un sistema con respecto al diseño, la implementación o los estándares de calidad que este cumple, en lugar de su comportamiento específico.

El desarrollo del sistema de información web para la gestión de la Vivienda Estudiantil y los becados residentes de la UNI, cumple con los siguientes requerimientos no funcionales:

6.2.1.2.3 Seguridad

El sistema cuenta con las siguientes características de seguridad:

- Autenticación mediante usuario y contraseña
- Implementación de estándares establecidos por el Proyecto Abierto de Seguridad de -Aplicaciones Web OWASP (por sus siglas en inglés, Open Web Application Security Project) para -verificación de fortaleza de contraseñas.
- Implementación de Certificados TLS
- Empleo de técnicas criptográficas.
- Registro de datos auditables para el monitoreo de las acciones realizadas por los usuarios.
- Empleo del certificado X.509 para autenticación de usuarios del lado del explorador web
- Prevención contra ataques XSRF
- Prevención contra ataques CSRF

6.2.1.2.4 Disponibilidad

El sistema tiene una disponibilidad de uso 100% las 24 horas del día, siempre y cuando estén funcionando los servidores y se tenga acceso a internet. El sistema no puede estar disponible cuando se presenten cortes de energía eléctrica que afecten los servidores que alojan la aplicación o exista algún problema en el proveedor de servicio de internet, cabe mencionar que estos, son inconvenientes causados por agentes externos, no por el sistema en sí.

6.2.1.2.5 Mantenibilidad

El IEEE1 (19990) define mantenibilidad como: “La facilidad con la que un sistema o componente software puede ser modificado para corregir fallos, mejorar su

funcionamiento u otros atributos o adaptarse a cambios en el entorno”. El sistema de información web para la Vivienda Estudiantil tiene una codificación modular, es decir que cada función es independiente una de la otra, esto permite corregir un fallo sin afectar la funcionalidad de otra opción. También permite agregar funcionalidades nuevas e integrarlas al sistema existente, debido al fácil uso y modificación del sistema.

La configuración del sistema estará parametrizada, es decir que un usuario capacitado puede ingresar al sistema y de manera rápida cambiar opciones, nuevos accesos, etc.

El administrador de la Vivienda Estudiantil es responsable de supervisar el uso diario del sistema y funcionamiento del sistema. En caso de que se presente una funcionalidad errada del sistema o algo está fallando, el administrador debe notificar al equipo que desarrolló el sistema para brindarle una atención y revisar las razones por las cuales se dan las fallas.

6.2.1.2.6 Portabilidad

El sistema para Vivienda Estudiantil es 100% portable, ya que está realizado en ambiente web, solo se necesita tener un navegador web instalado en una máquina o teléfono inteligente, y conectarse a la internet y el sistema queda totalmente funcional.

Si se necesita tener acceso al sistema desde cualquier punto, se necesita subir totalmente a la nube, o habilitar una red privada virtual (vpn) para el acceso. Los datos son 100% dependientes del servidor de base de datos, así como también el sitio web del servidor web.

Es preciso destacar que, en la interfaz de usuario del sistema, se utilizó diseño responsivo, lo que permite acceder al mismo desde diversos dispositivos móviles con diferentes resoluciones de pantalla.

A continuación, se muestran algunos de los dispositivos en los cuales se realizaron pruebas del sistema.

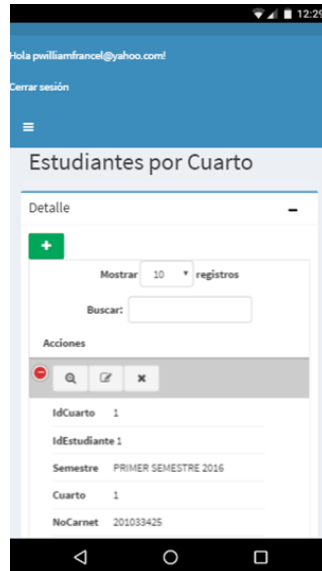


Figura 5: Interfaz del sistema, teléfono inteligente Nexus 5x con resolución de pantalla de 412 x 732 pixeles (Elaboración propia)

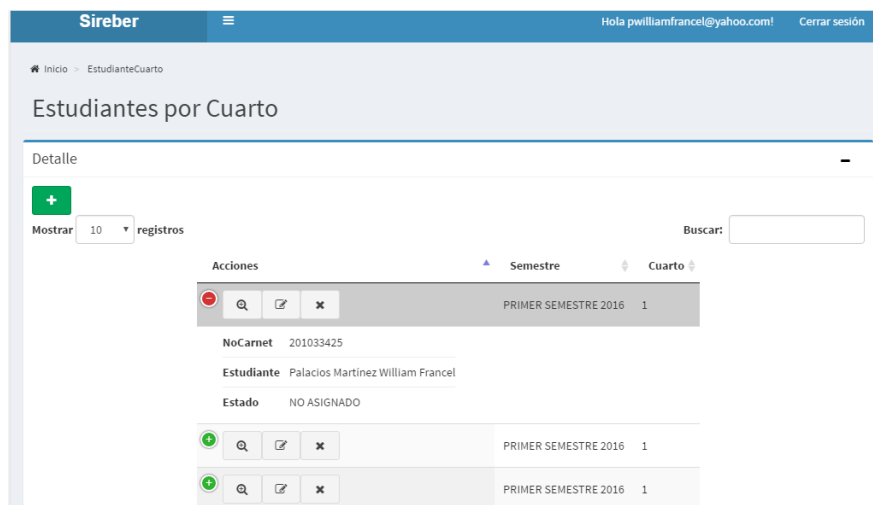


Figura 6: Interfaz del sistema, Ipad con resolución de pantalla de 1024 x 768 pixeles.

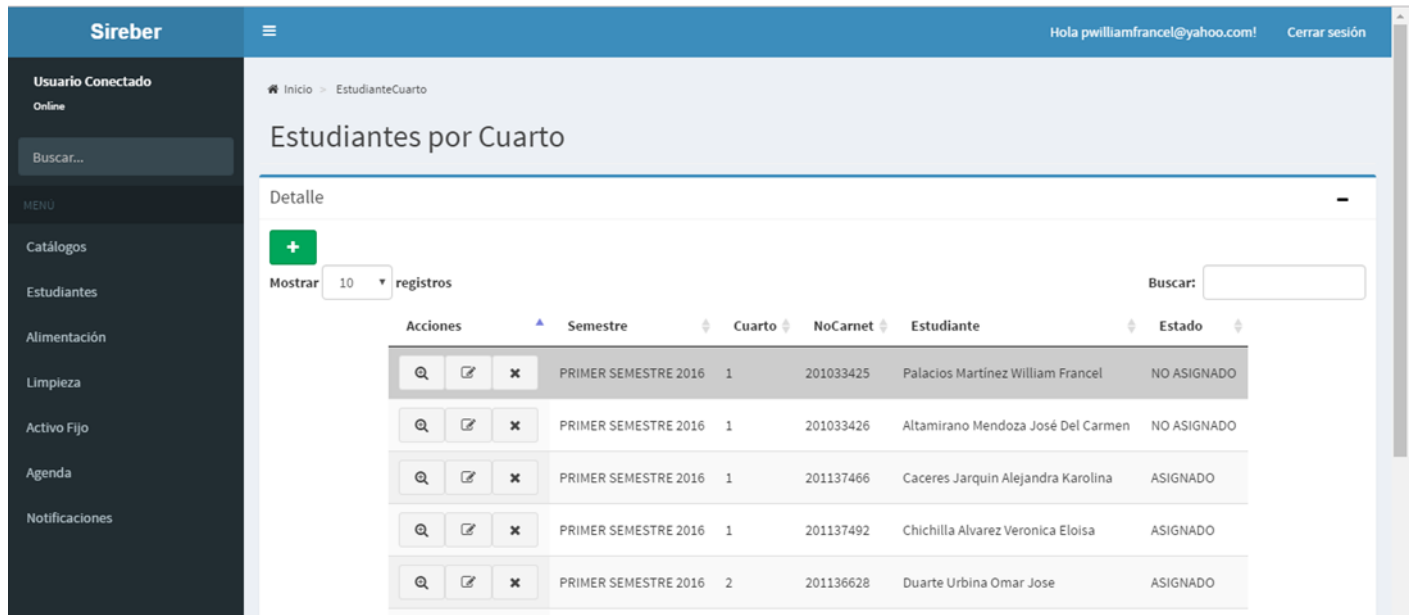


Figura 7: Interfaz del sistema desde una PC con resolución de pantalla de 1366 x 768 pixeles.

6.2.1.2.7 Restricciones

Como producto del análisis de requerimientos funcionales y no funcionales, se delimitaron las siguientes restricciones:

- El sistema solo se ocupa de los procesos administrativos de la Vivienda Estudiantil.
- En cuanto a la tecnología, se trabajará únicamente con el gestor de base de datos SQL server 2008 R2 o una versión superior, en conjunto con la plataforma ASP MVC 5 del framework .Net en su versión 4.5 o superior.
- El sistema web, solamente podrá ser publicado en un servidor Windows versión 2012 o superior.

- Se accederá al sistema, únicamente utilizando los siguientes navegadores web: Google Chrome versión 41 o superior, Mozilla Firefox 39 o superior, Internet Explorer 11 o superior, Microsoft Edge 25 o superior y Safari 9 o superior.
- El sistema no tiene opción de interactuar con otro sistema externo, ya que está diseñado para trabajar de manera independiente, para lograr esta característica, se deberá pasar por el debido proceso de desarrollo e integración.
- El módulo de alimentación, en la función de control de entrega de alimentos, evita que se entreguen órdenes de comida de más, siempre y cuando el encargado del comedor haga uso del método de validación que incluye el sistema, que es por medio de verificación de carnet estudiantil.
- Para el funcionamiento satisfactorio del sistema de información web de la Vivienda Estudiantil debe de estar conectado a una red de internet.
- El sistema realiza los procesos de forma adecuada, siempre y cuando los usuarios ingresen correctamente los datos.
- El sistema tiene una disponibilidad de uso 24 horas al día y 7 días a la semana; siempre y cuando los servidores de base de datos y web donde esté alojado el sistema, estén en funcionamiento, para garantizar un acceso constante a la aplicación.

6.2.1.2.8 Suposiciones y dependencias

Factores que pueden afectar los requerimientos del sistema:

- No utilizar exploradores web conocidos para acceder al sistema (por ejemplo, Google Chrome, Mozilla Firefox, Safari, Internet Explorer y Microsoft Edge)
- Cambiar en el proceso actual de administración de la Vivienda Estudiantil.

- Utilizar servidores que no cumplan con los requisitos de hardware y software especificados en la factibilidad técnica (**Ver Anexo 3: Factibilidad Técnica**).
- Falta de acceso a internet.
- Falta de acceso a la información pertinente de los estudiantes y a la administración de la Vivienda Estudiantil.
- Desactualización de la información de los estudiantes becados residentes

6.2.1.3 Sprint #3 – Diseño de Base de Datos e Interfaz Gráfica del Sistema

6.2.1.3.1 Diseño de Base de Datos

Una vez definidos los requerimientos funcionales y no funcionales, se procedió a diseñar el modelo de datos que utilizaría el sistema, mediante la implementación de la Metodología de Diseño de Bases de Datos Relacionales, la cual plantea el siguiente flujo:

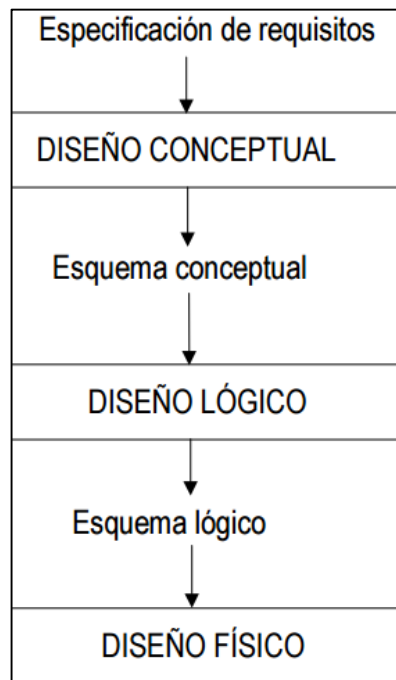


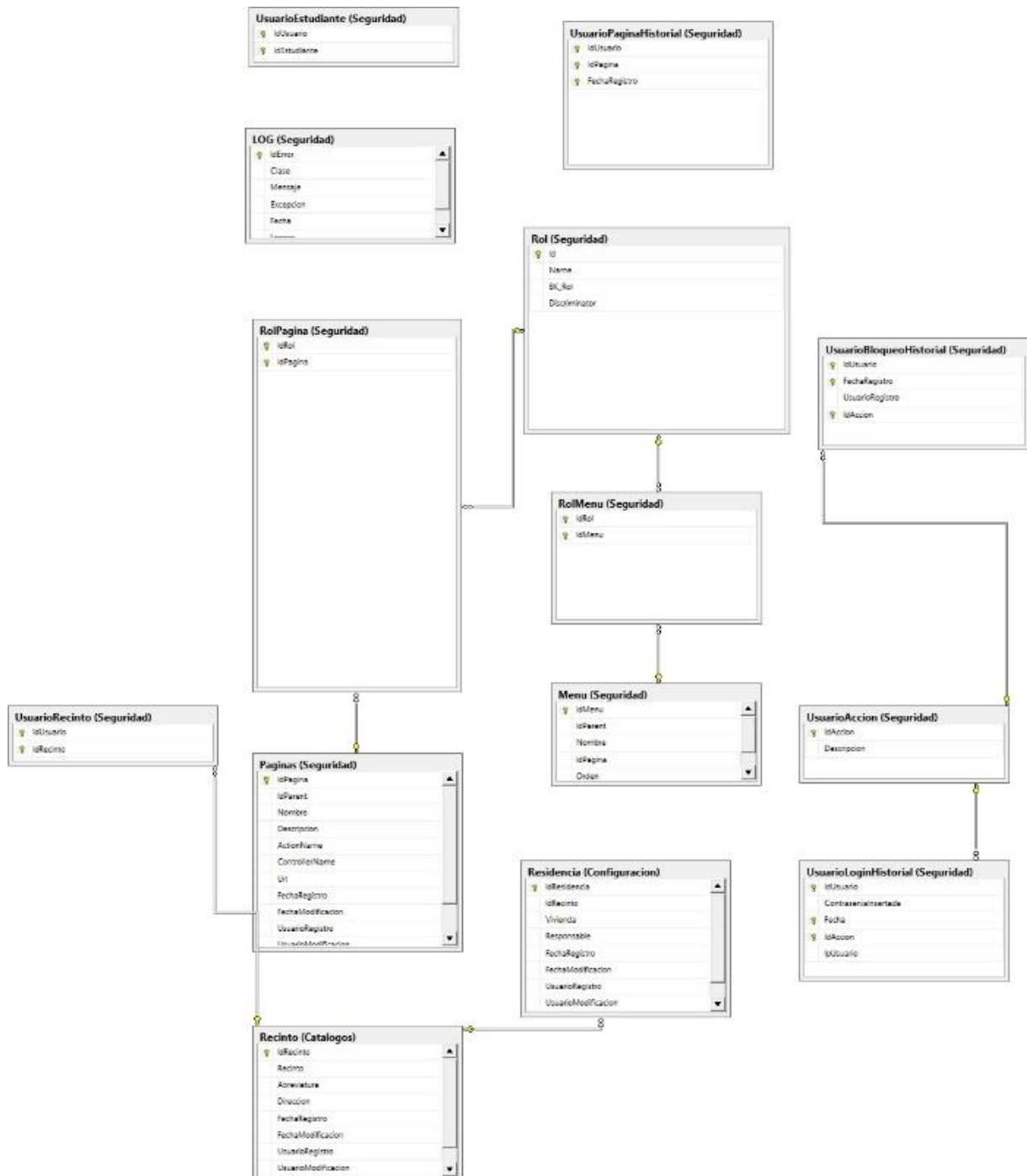
Figura 8: Flujo diseño de base de datos (Elaboración propia)

Inicialmente, se aplicó el del diseño conceptual de la base de datos, esto es, identificar, por medio de los requerimientos funcionales definidos, las entidades y relaciones correspondientes a la realidad.

Luego, se realizó el diseño lógico, proceso en el cual, se transforman las entidades definidas, a objetos más amigables a los gestores de bases de datos, mediante el uso de técnicas de normalización (en este caso, se aplicó hasta la tercera forma normal) para garantizar la consistencia y evitar la redundancia de los datos.

En esta etapa de diseño, se obtuvo como resultado, el modelo relacional, el cual fue implementado en el sistema gestor de base de datos SQL Server Management Studio 2008 R2, este gestor fue seleccionado debido a los factores tomados en cuenta en la factibilidad técnica (**Ver Anexo 3: Factibilidad Técnica**).

El modelo de datos relacional, resultante del proceso de diseño se muestra a continuación:



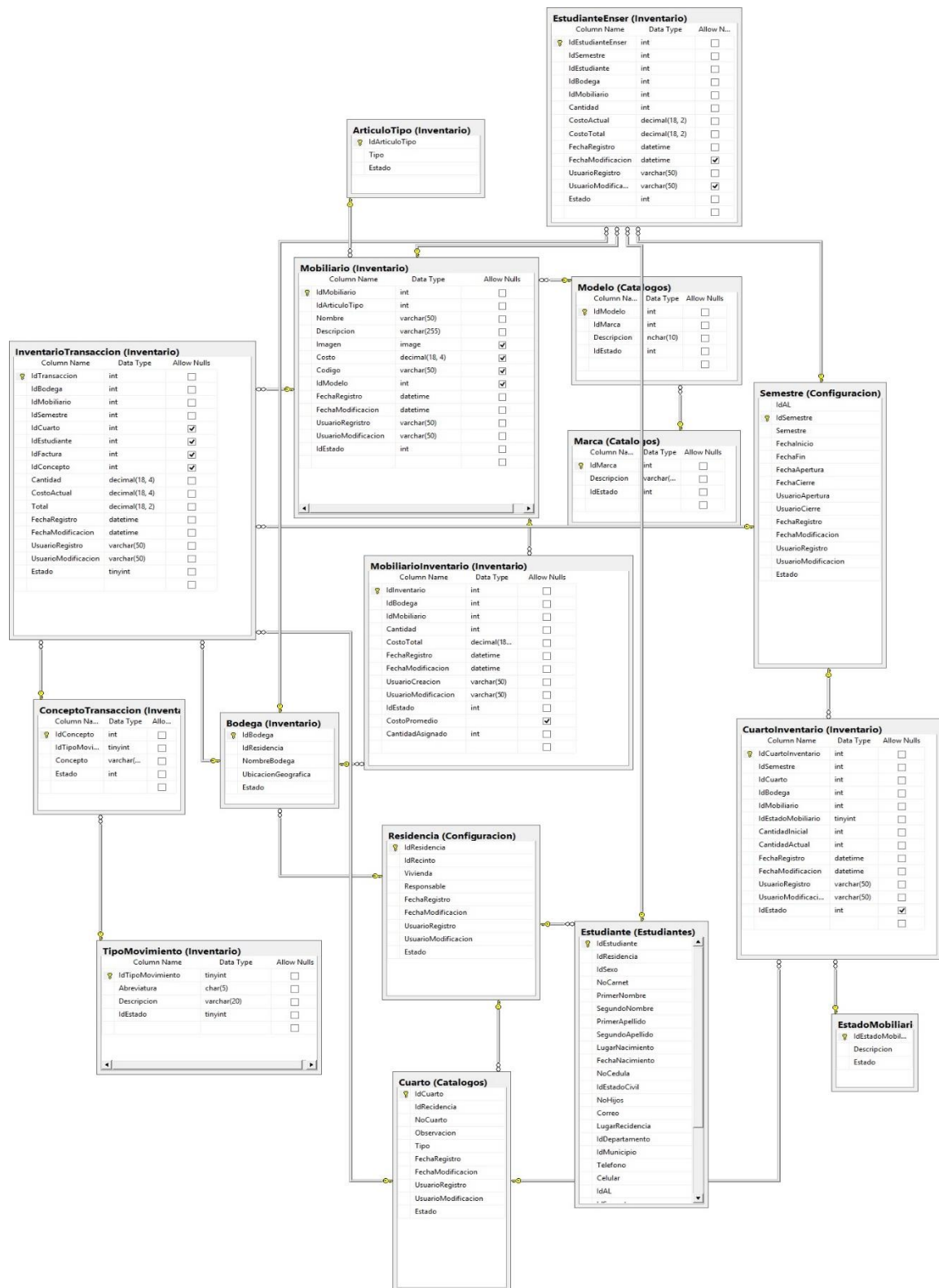


Figura 9: Muestra diagrama Entidad-Relación (ERD)

6.2.1.3.2 Diseño de Interfaz Gráfica del Sistema

Un diseño cuidadoso de la interfaz de usuario es parte fundamental del diseño general del software, esta debe ajustarse a las habilidades, experiencia y expectativas de los usuarios que la utilizarán. El buen diseño de la interfaz de usuario es crítico para la confiabilidad del sistema, muchos de los llamados “errores de usuario” son causados por el hecho de que las interfaces de usuario no consideran las habilidades de sus usuarios reales y su entorno de trabajo.

Partiendo de estas premisas, la interfaz de usuario del sistema de información web para la gestión de la Vivienda Estudiantil y becados residentes de la UNI, fue diseñada aplicando los siguientes principios:

- Familiaridad del Usuario. La interfaz debe utilizar términos y conceptos obtenidos de las experiencias de las personas en el negocio u otros sistemas utilizados.
- Uniformidad de la Interfaz de Usuario: Siempre que sea posible, la interfaz debe ser uniforme, en el sentido en que las operaciones comparables se activen de la misma forma.
- Mínima Sorpresa: el comportamiento del sistema no debe generar sorpresas a los usuarios.
- Recuperabilidad: La interfaz debe incluir mecanismos para permitir a los usuarios recuperarse de los errores.
- Diversidad de Usuario: La interfaz debe proporcionar características de interacción apropiadas para los diferentes tipos de usuarios del sistema.
- Reducción de la Carga de Memoria al Usuario: Se debe de evitar, en la medida de lo posible, utilizar interfaces de usuario que contengan

demasiados elementos y que compliquen la capacidad de memorización del usuario.

- Diseño adaptativo: La interfaz de usuario debe ser capaz de adaptar su apariencia con respecto a la pantalla del dispositivo que se esté utilizando para visualizarlas (por ejemplo: tabletas, teléfonos inteligentes, computadoras portátiles y de escritorio).
- Ergonomía: ISO/TS 16071 – Ergonomía de la Interacción hombre-máquina, plantea que, cualquier sistema que pueda ser objeto de interacción humana, debe ser confortable y de fácil uso (simplicidad, claridad y facilidad de comprensión), para mejorar la calidad de vida del trabajador que opera el sistema.

Mediante el cumplimiento de los principios mencionados y la aplicación del estándar HTML ISO-8859-1, se diseñaron interfaces de formularios (utilizando HTML5, CSS3 y JavaScript), también conocidas como interfaces de entrada y salida, las cuales, constan de campos que contienen datos o parámetros que necesitan ser comunicados al usuario.

La figura 10, 11 y 12 muestran interfaces de formulario utilizadas para registrar y listar horarios de alimentación respectivamente.

The screenshot shows a web application interface for 'Horario de Alimentación'. The title bar includes 'Inicio' and 'HorarioAlimenticio'. The main heading 'Horario de Alimentación' is highlighted with a red box. Below it, a 'Detalle' section contains several input fields: 'Código de Horario' (text box with '0'), 'Recinto' (dropdown menu with '-- Seleccionar Recinto --'), 'Nombre' (text box), 'Hora de Inicio' (time picker set to '12:15 PM'), 'Hora de Finalización' (time picker set to '12:15 PM'), and 'IdEstado' (toggle switch labeled 'INACTIVO'). At the bottom are 'Guardar' and 'Cancelar' buttons. Annotations with red lines point to the title bar and the form fields, accompanied by descriptive text boxes.

Familiaridad del Usuario, todas las interfaces de usuario contienen descripciones, con términos aplicados en el negocio

Uniformidad de la Interfaz de Usuario, las interfaces del sistema, utilizan la misma posición, márgenes, colores, texturas y controles para los campos de entrada y muestra de datos.

Diversidad de Usuario: Interfaces intuitivas que facilitan el uso del sistema para diversos tipos de usuario.

Figura 10: Interfaz de Formulario para Registro de Horarios de Alimentación

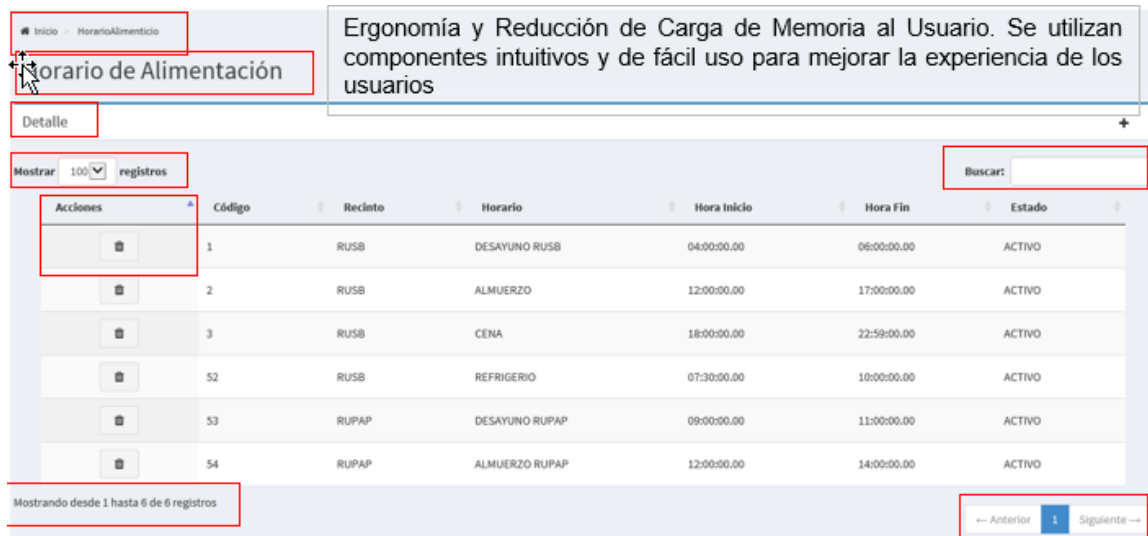


Figura 11: Interfaz de Formulario para Listar Horarios de Alimentación

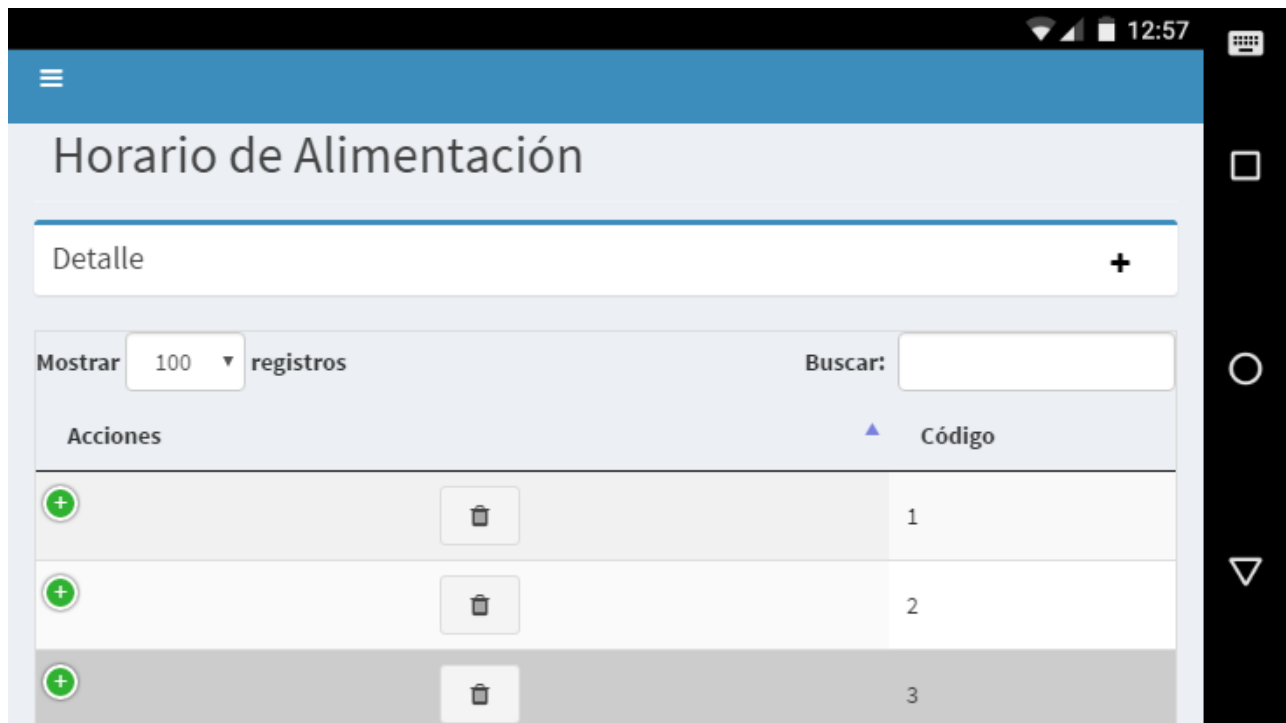


Figura 12: Interfaz de Formulario con diseño adaptativo para Listar Horarios de Alimentación (vista desde un teléfono Nexus 6 P)

6.2.1.3.3 Navegabilidad del Sistema

La navegabilidad es la facilidad con la que un usuario puede desplazarse por todas las páginas que componen un sistema. Para lograr este objetivo, un sitio web debe proporcionar un conjunto de recursos y estrategias de navegación diseñados para conseguir un resultado óptimo en la localización de la información y en la orientación para el usuario. En el sistema de información web para la gestión de la Vivienda Estudiantil y becados residentes de la UNI, se utilizó el tipo de navegabilidad Jerárquica.

La estructura jerárquica, es una estructura de árbol, en el que la raíz es la página de bienvenida, esta página se puede también sustituir por otra de contenido, en la que se exponen las diferentes secciones que contendrá el sitio. La selección de una sección conduce al usuario a una lista de subíndices que pueden o no dividirse.

Este tipo de organización permite al usuario conocer en qué lugar de la estructura se encuentra, además de saber que, conforme se adentra en la estructura obtiene información más específica y que la general se encuentra en los niveles superiores, a como se muestra a continuación:

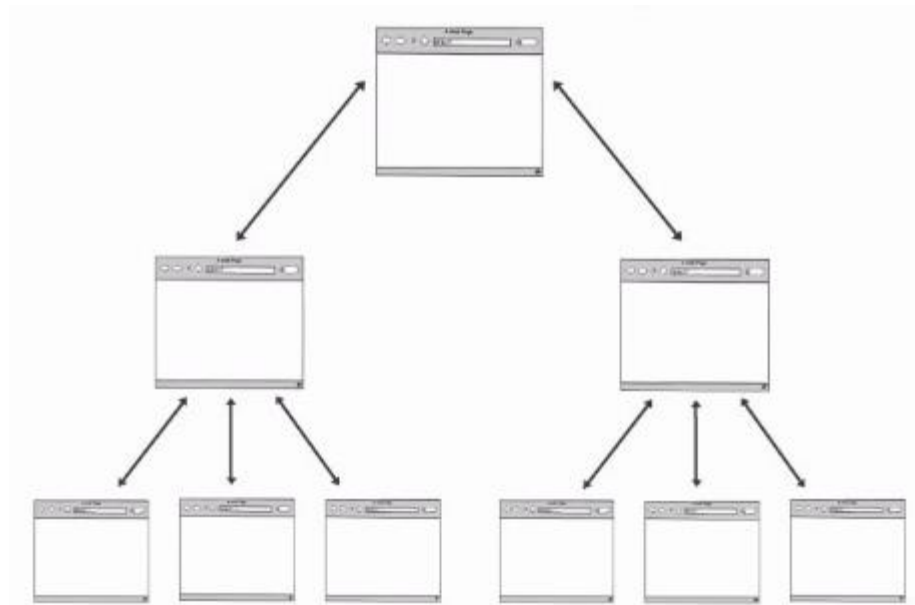


Figura 13: Navegabilidad Jerárquica del sistema

6.2.1.4 Sprint #4 – Diseño de la Arquitectura del Sistema

La guía de Microsoft (2015) para el diseño de arquitecturas de aplicaciones, define el diseño de la arquitectura del software como “el proceso de definición de una solución estructurada, que cumple con todos los requisitos técnicos y operativos, optimizando al mismo tiempo, los atributos de calidad, como el rendimiento, la seguridad y facilidad de administración”.

Al igual que cualquier otra estructura compleja, el software debe ser construido sobre una base sólida, se deben de tener en cuenta principios claves, para apreciar las consecuencias a largo plazo, con la consideración para el usuario y los objetivos del negocio.

En el proceso de diseño de la arquitectura del sistema de información web para la gestión de la Vivienda Estudiantil y becados residentes de la UNI se llevó a cabo con los siguientes pasos:

- Selección de los principios de diseño de arquitectura de software
- Selección e Implementación de Patrones Arquitectónicos

6.2.1.4.1 Principios de Diseño de Arquitectura de Software

Los principios de diseño de la arquitectura de software son fundamentales porque ayudan a crear una arquitectura que minimiza costos y requisitos de mantenimiento y promueven la facilidad de uso y extensibilidad.

Los principios tomados en cuenta en el sistema de información web para la gestión de la Vivienda Estudiantil y becados residentes de la UNI son:

- Separación de Intereses
- Responsabilidad Simple
- No te Repitas (DRY por sus siglas en inglés, Don't Repeat Yourself)
- Mantener los patrones de diseño coherentes en cada capa de la aplicación
- No duplicar código a lo largo de la aplicación
- Establecer un estilo de codificación y convenciones de nombres para desarrollo.

Patrones Arquitectónicos del Sistema

Un patrón arquitectónico, es un conjunto de principios que proporciona un marco abstracto para una familia de sistemas, es decir, son un conjunto de principios que dan forma a una aplicación, mejoran la separación y promueven la reutilización del diseño.

A continuación, se describen los patrones arquitectónicos seleccionados e implementados en el sistema:

- Diseño Orientado a Dominio (DDD por sus siglas en inglés, Domain Driven Design)
- Patrón arquitectónico de capas (Layered Architectural Pattern)

Diseño Orientado a Dominio

Es un enfoque orientado a objetos para el diseño de software, basado en el dominio del negocio, sus elementos y comportamientos y las relaciones entre ellos. Su objetivo es permitir que los sistemas de software interpreten el negocio, mediante la definición de un modelo, también conocido como entidad, que se expresa en el lenguaje de programación determinado.

Los principales beneficios de este patrón son:

- **Comunicación:** Todas las partes dentro de un equipo de desarrollo pueden utilizar el modelo de dominio y las entidades que define para comunicar conocimientos y requisitos de negocio, utilizando un lenguaje de dominio de la empresa común, sin que se requiera la jerga técnica.
- **Extensible:** El modelo de dominio a menudo es modular y flexible, por lo que es fácil de actualizar y ampliar las condiciones y requisitos cambian.
- **Comprobable:** Los objetos del modelo de dominio están unidos ligeramente y cohesionada, lo que permite su aprobación más fácilmente.

Patrón Arquitectónico de Capas

La arquitectura en capas se centra en la agrupación de funcionalidad relacionada dentro de una aplicación, en distintas capas que se apilan verticalmente una encima de la otra, estas capas ayudan a mantener una fuerte separación de intereses que, a su vez, apoya la flexibilidad y facilidad de mantenimiento.

La funcionalidad dentro de cada capa está relacionada con un papel o responsabilidad común. La comunicación entre capas es explícita y débilmente acoplada.

Algunos principios que usa el patrón arquitectónico de capas son:

- Abstracción
- Encapsulamiento
- Alta Cohesión
- Reusabilidad
- Bajo Acoplamiento
- Separación de Intereses
- Alto Desempeño
- Manejabilidad
- Escalabilidad
- Mantenibilidad

Como resultado de la selección de patrones de diseño, la arquitectura final del sistema se muestra en la figura 14, esta fue implementada en la plataforma .Net,

con el patrón MVC 5 y el lenguaje de programación C#, a como se muestra en la figura 15 y 16.

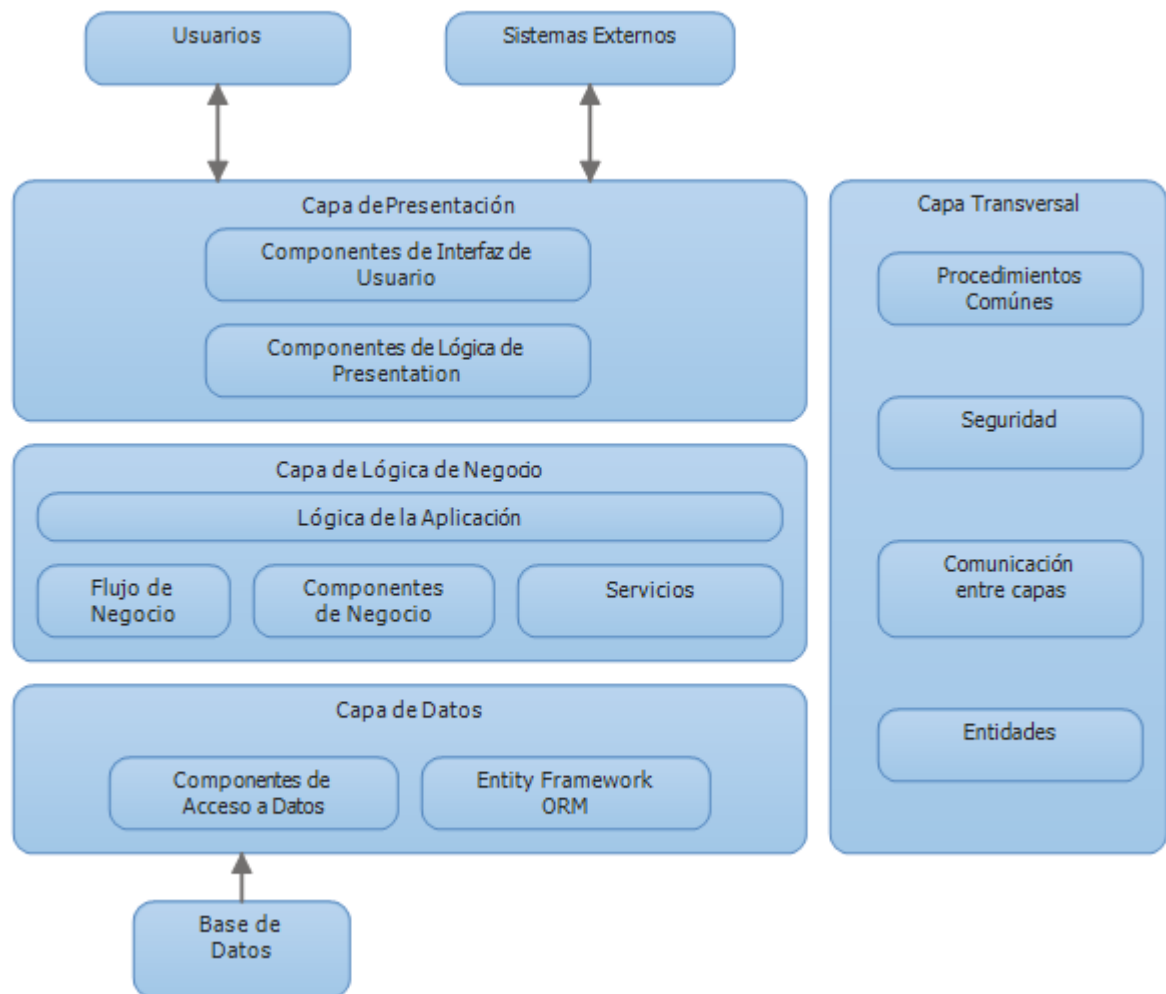


Figura 14: Arquitectura de Software (Elaboración propia)

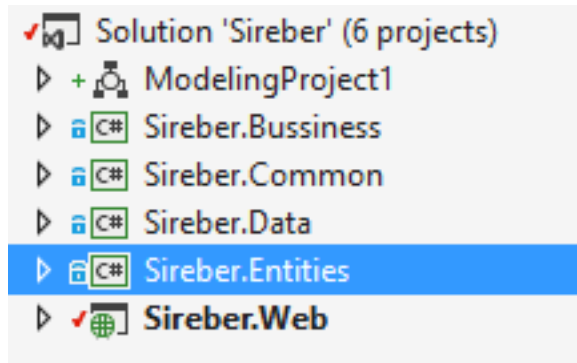


Figura 15: Solución de la aplicación en .Net (Elaboración propia)

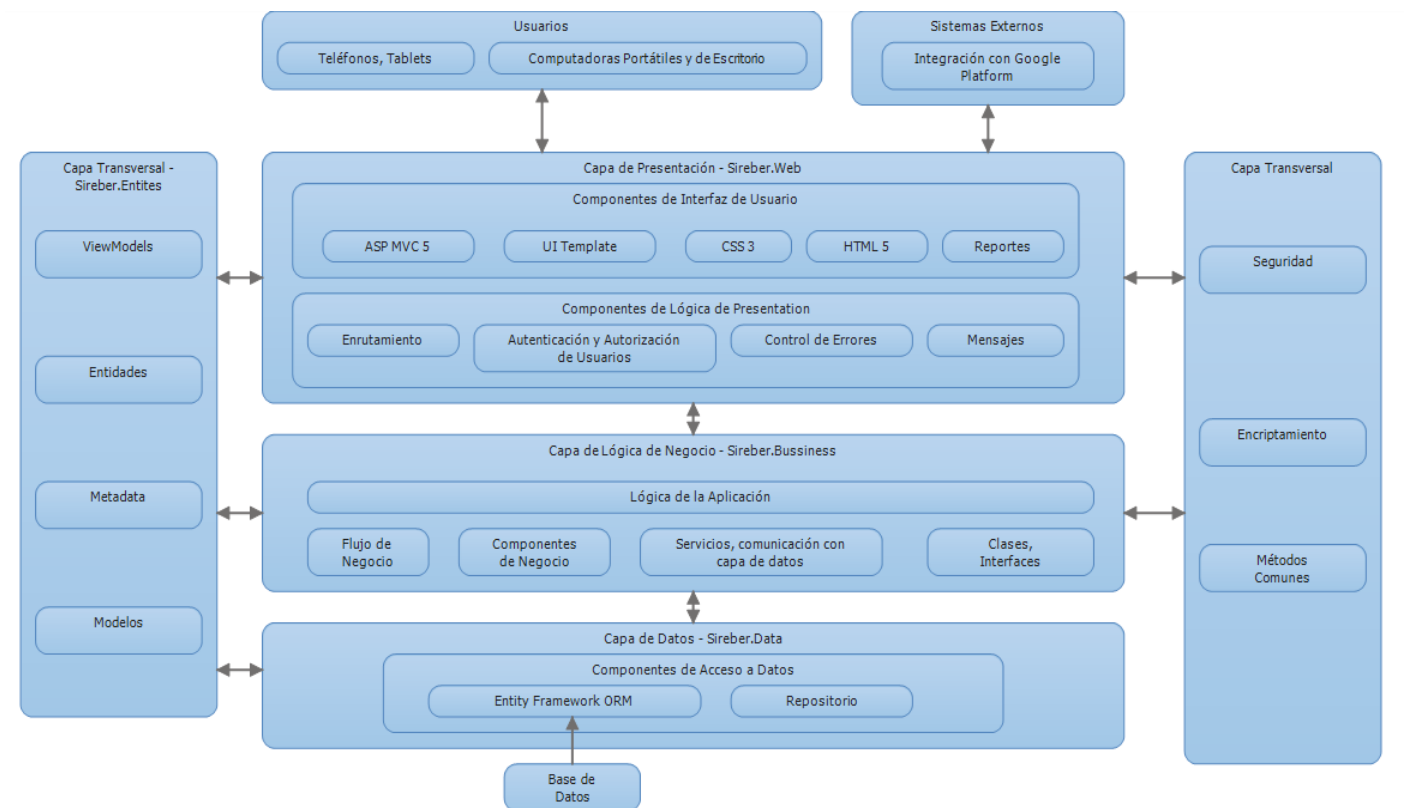


Figura 16: Arquitectura de Capas y DDD Aplicada a la solución del Sistema (Elaboración propia)

6.2.1.5 Sprint #5-12: Desarrollo de Módulos Sistema de Información web para la gestión de la Vivienda Estudiantil

A continuación, se describe los pasos que se dieron en el desarrollo de cada módulo y su codificación.

6.2.1.5.1 Estructura física del sistema de información web para la gestión de la Vivienda Estudiantil y los becados residentes

El siguiente esquema representa cómo está constituida la estructura física del sistema de información web para la Vivienda Estudiantil

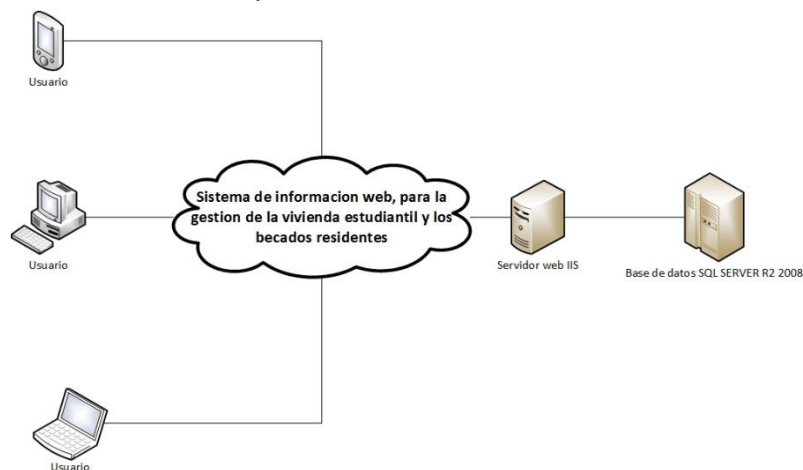


Figura 17: Estructura del sistema de información web para la vivienda estudiantil (Elaboración propia)

El sistema almacena los datos en un servidor SQL SERVER R2, y la aplicación web se aloja en el servidor web IIS de Microsoft. Una vez instalados ambos ambientes, los clientes podrán acceder al sistema de información de la Vivienda Estudiantil a través de internet.

6.2.1.5.2 Diseño lógico del sistema de información web para la gestión de la Vivienda Estudiantil y los residentes becados

Una vez definida la estructura, se realizó el diseño lógico del software para la administración de la Vivienda Estudiantil con base en requerimientos analizados en conjunto con el cliente. Los diseños que se muestran representan la lógica de los procesos administrativos llevados a cabo en la Vivienda Estudiantil, los diagramas permitirán al usuario una mejor comprensión de los requerimientos y funcionalidad del sistema.

En el documento se utilizó diseño de alto nivel o arquitectónicos, representando así cada módulo del sistema.

Se elaboraron varios diagramas. En este acápite del documento se presentan los más importantes (para ver los demás diagramas **ver Anexo 12: Diagramas lenguaje unificado de modelado (UML) para el diseño lógico del sistema de información web para la Vivienda Estudiantil**).

A continuación se presentan los diagramas de diseño con base en las plantillas de casos de uso, Todos los diagramas son elaboración propia. (**Ver Anexo 11: Plantillas de caso de uso**).

6.2.1.5.3 Diagramas de flujo

Módulo inventario

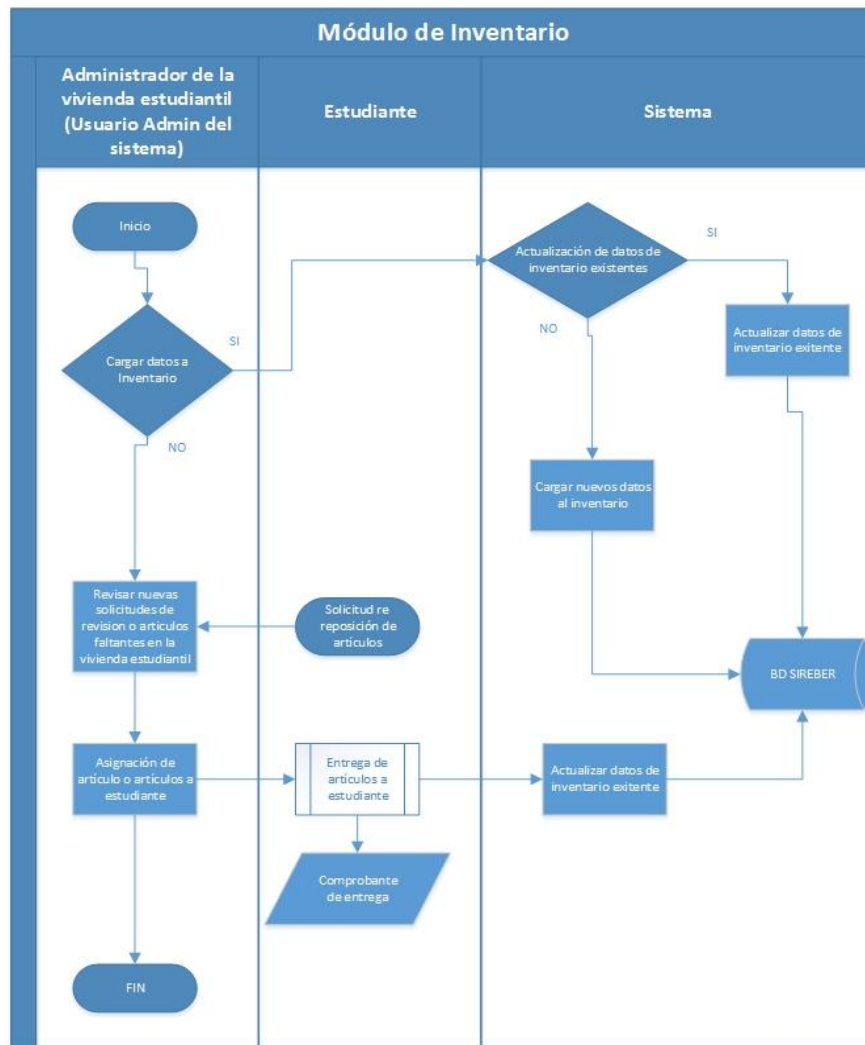


Figura 18: Diagrama de flujo Modulo de Inventario

Módulo de alimentación

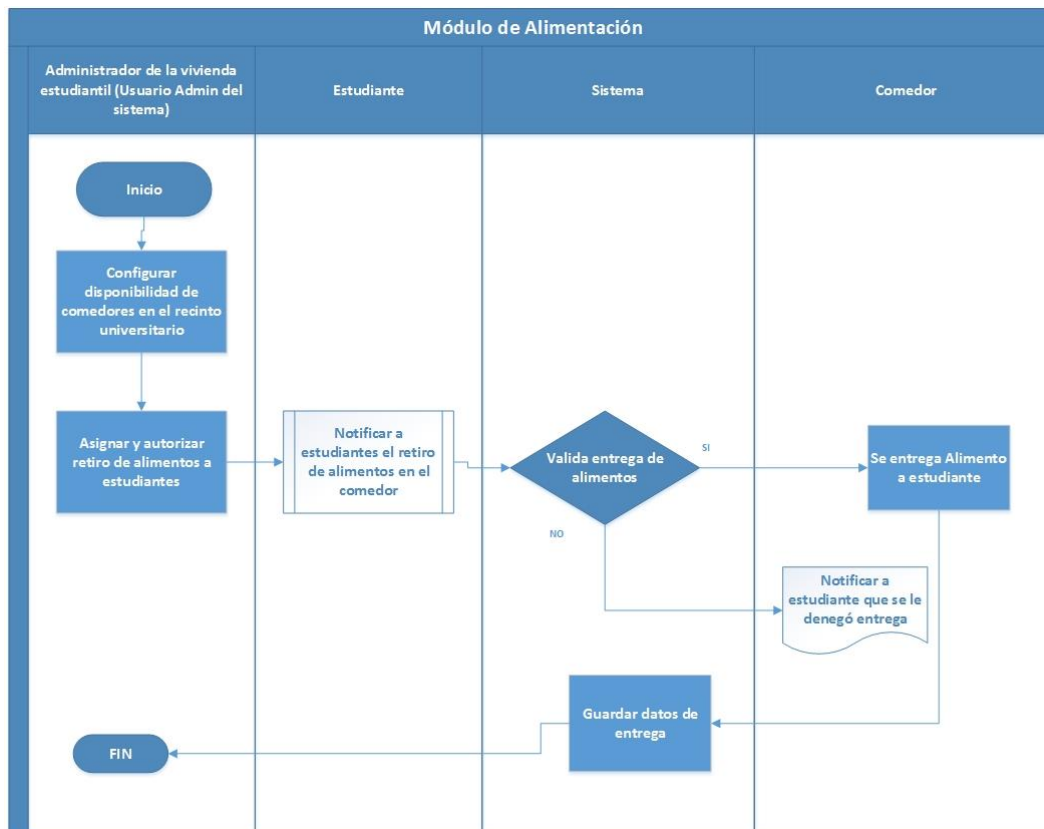


Figura 19: Diagrama de flujo módulo de Alimentación

Módulo de Creación actividad social

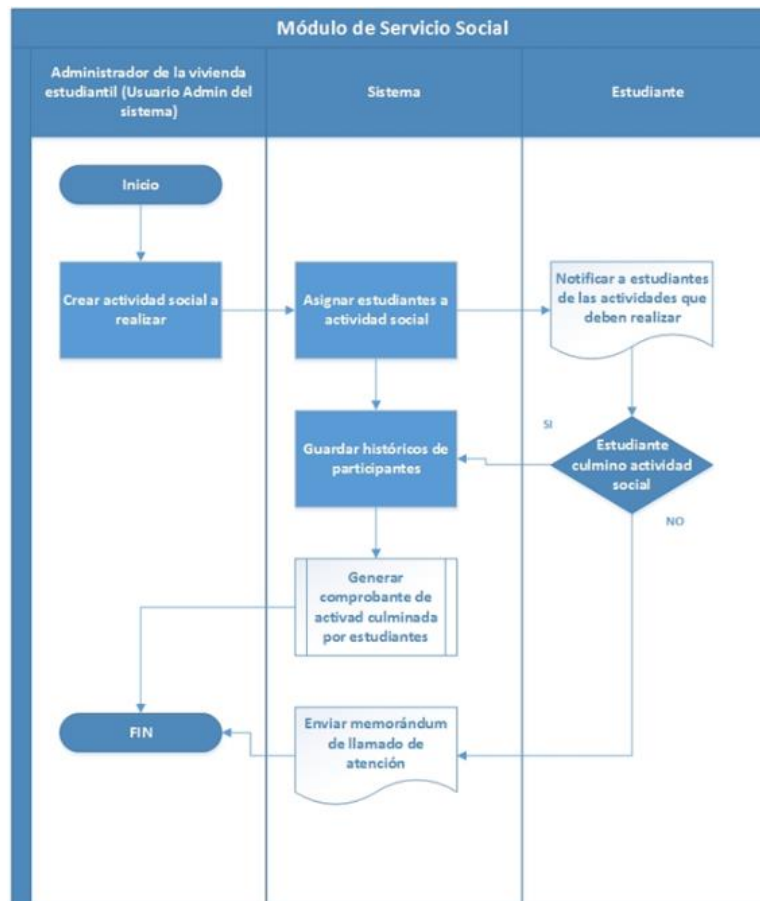


Figura 20: Diagrama de flujo módulo de creación de actividad social

6.2.1.5.4 Diagramas de caso de uso

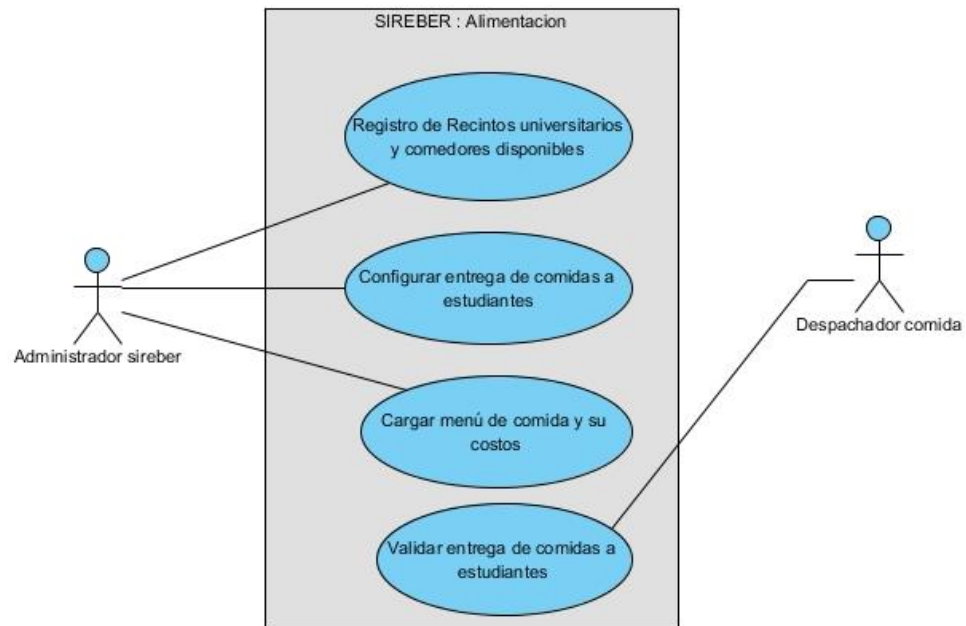


Diagrama 1: Módulo de alimentación

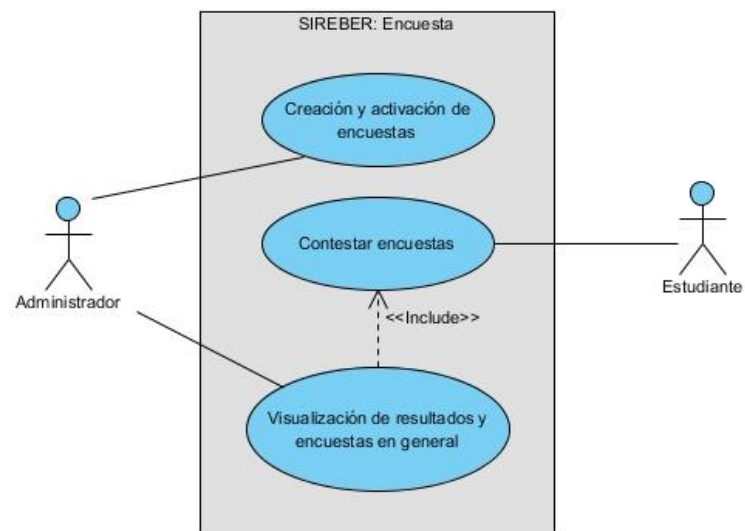


Diagrama 2: Módulo de encuestas

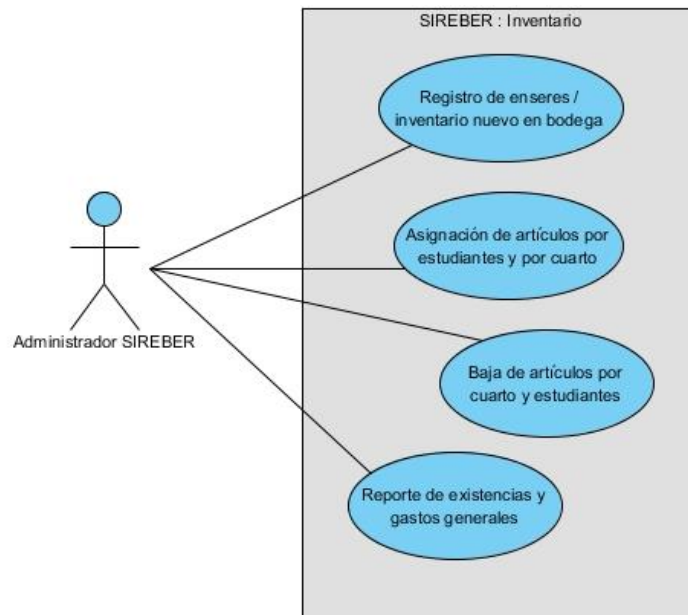


Diagrama 3: Módulo Inventario

6.2.1.5.5 Diagramas de actividad

Diagrama de actividad asociado a caso de uso: Registro de enseres / inventario nuevo en bodega, actor: Administrador.

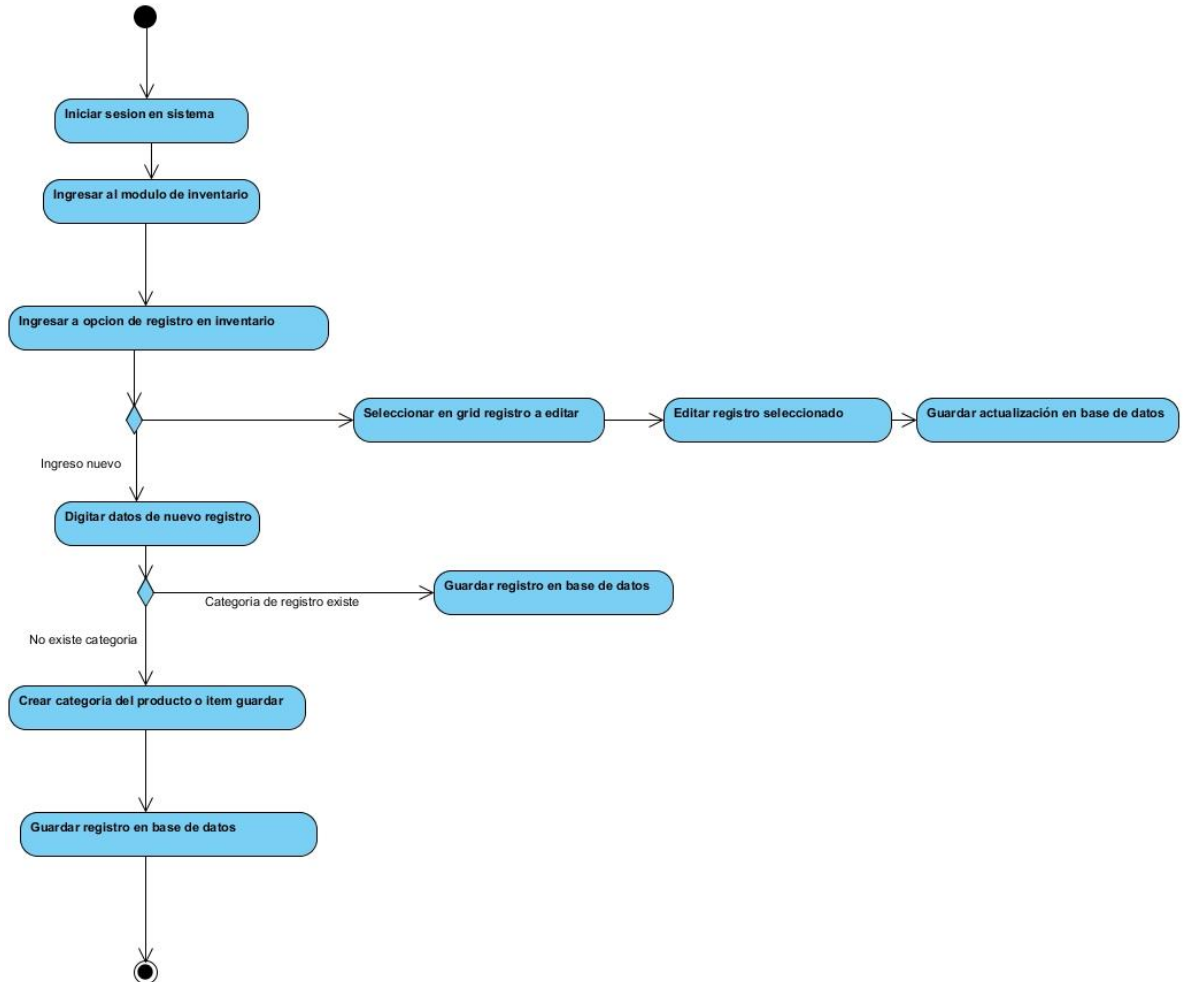


Diagrama 4: Proceso Registro de enseres / Inventario Nuevo

Diagrama de actividad asociado a caso de uso: Asignación de artículos por estudiantes y por cuarto, actor: Administrador.

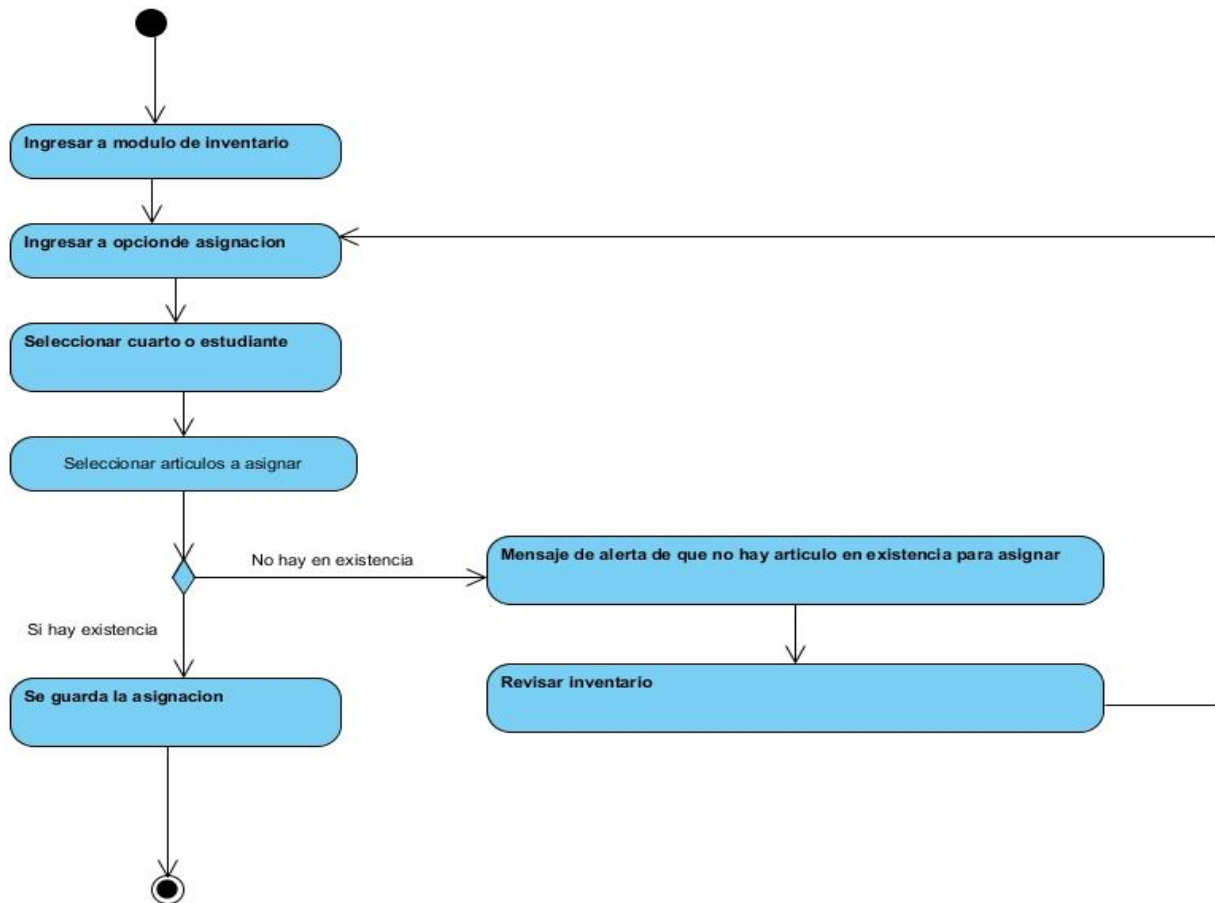


Diagrama 5: Asignación de artículos

Diagrama de actividad asociado acaso de uso: Registro de Recintos universitarios y comedores disponibles, actor: Administrador.

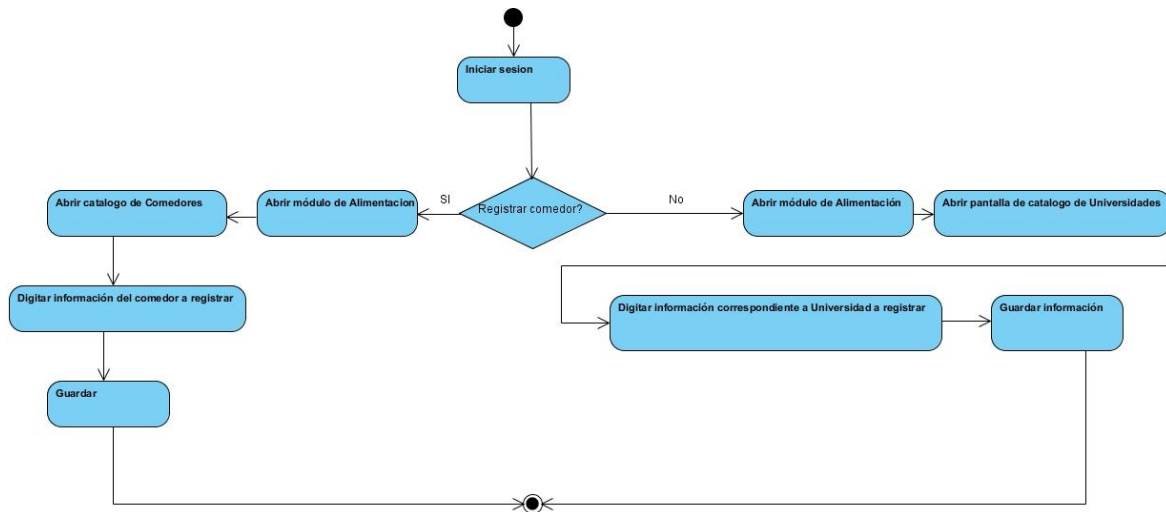


Diagrama 6: Proceso registro de Universidad o comedor

Diagrama de actividad asociado acaso de uso: Validar entrega de comidas a estudiantes mediante Número De carnet, actor: Despachador comedor, estudiante

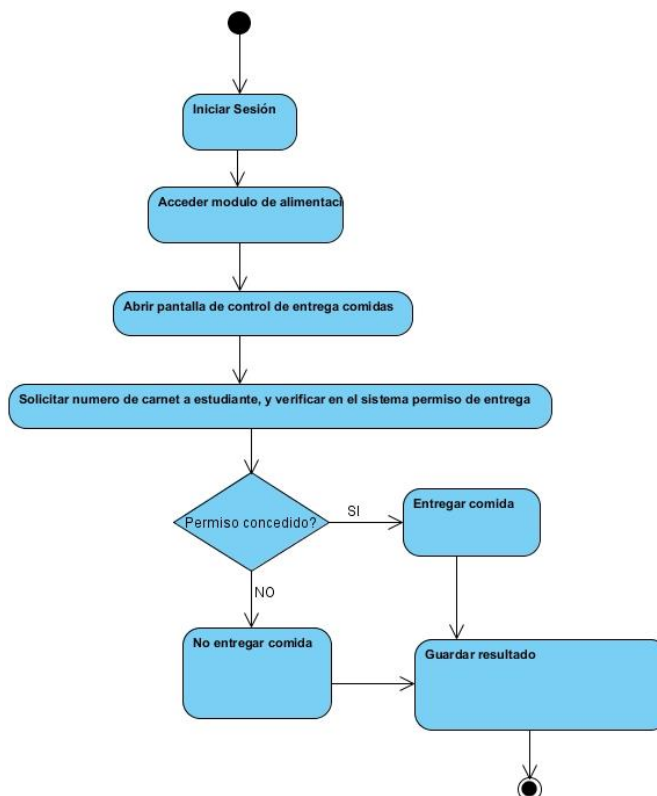


Diagrama 7: Validar entrega de comidas a estudiantes mediante Número De carnet

Diagrama de actividad asociado a caso de uso: Generar Rol de aseo, actor:
Administrador

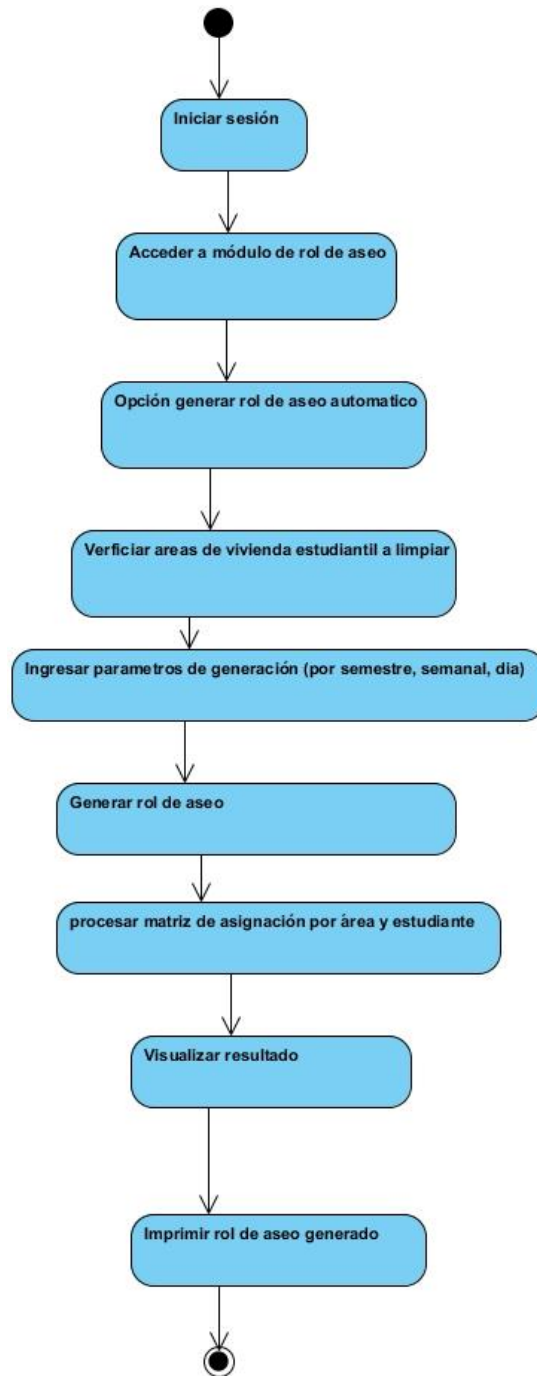


Diagrama 8: Proceso Generar rol de aseo

6.2.1.5.6 Diagramas de secuencia

Diagrama de secuencia asociado a caso de uso: Asignación de artículos por estudiantes y por cuarto, actor: Administrador

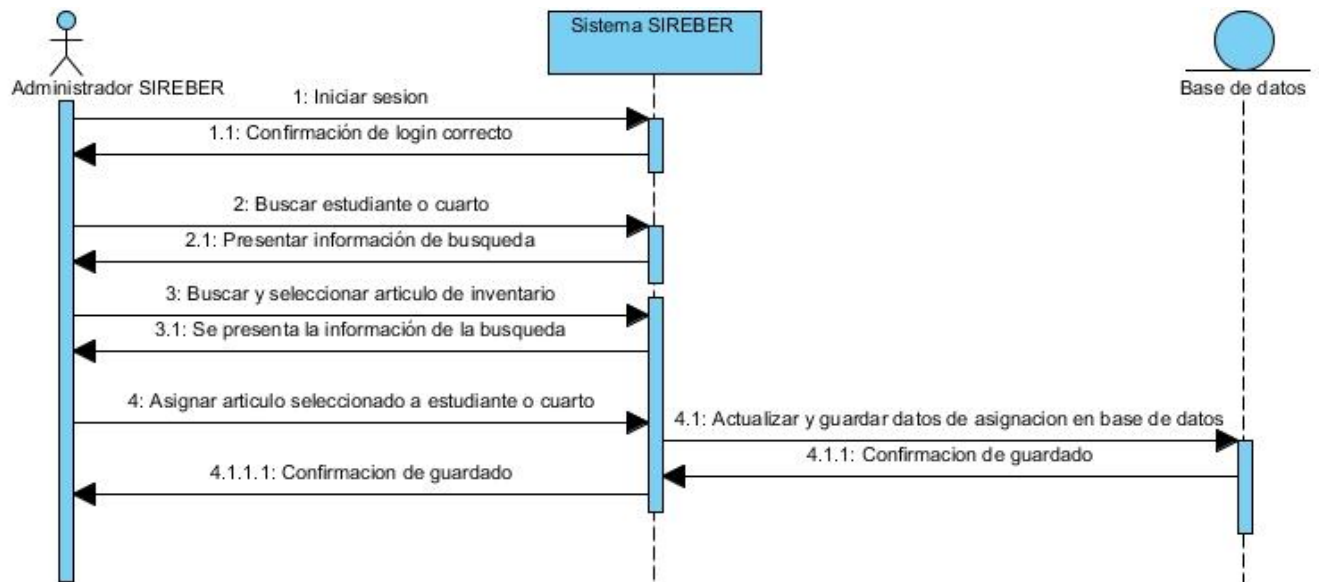


Diagrama 9: Proceso de asignación de artículos de inventario

Diagrama de secuencia asociado a caso de uso: Baja de artículos por cuarto y estudiantes, actor: Administrador

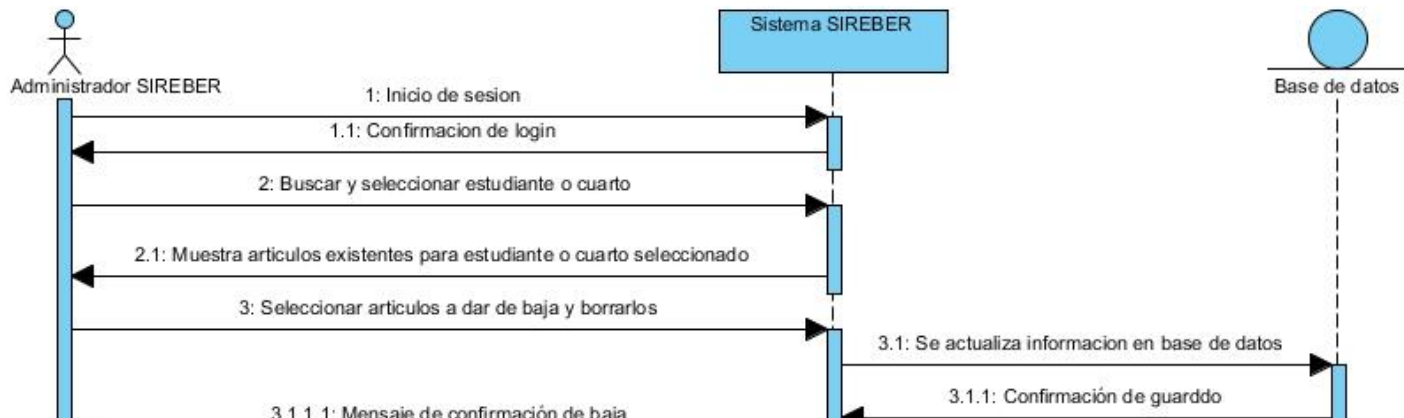


Diagrama 10: Proceso de bajas en inventario

Diagrama de secuencia asociado a caso de uso: Configurar entrega de comidas a estudiantes mediante lector biométrico o carnet estudiantil en los comedores, actor: Administrador

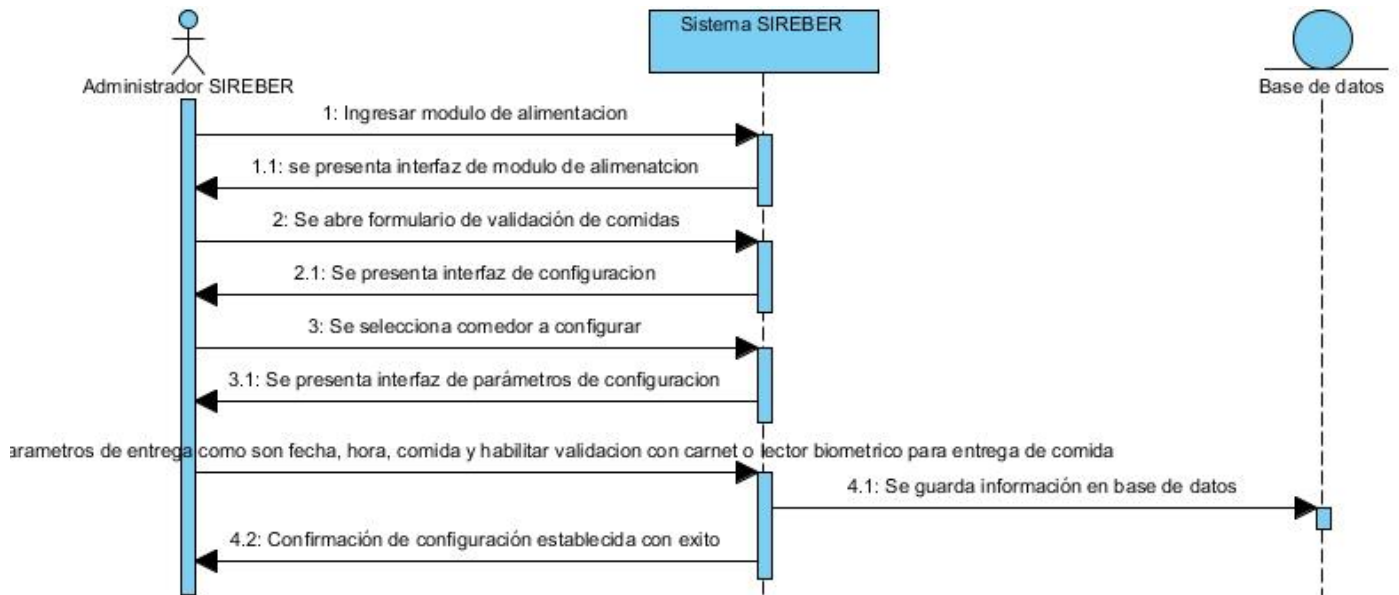


Diagrama 11: Proceso de configuración de entrega de comidas

Diagrama de secuencia asociado a caso de uso: Validar entrega de comidas a estudiantes mediante Número De carnet, actor: Administrador

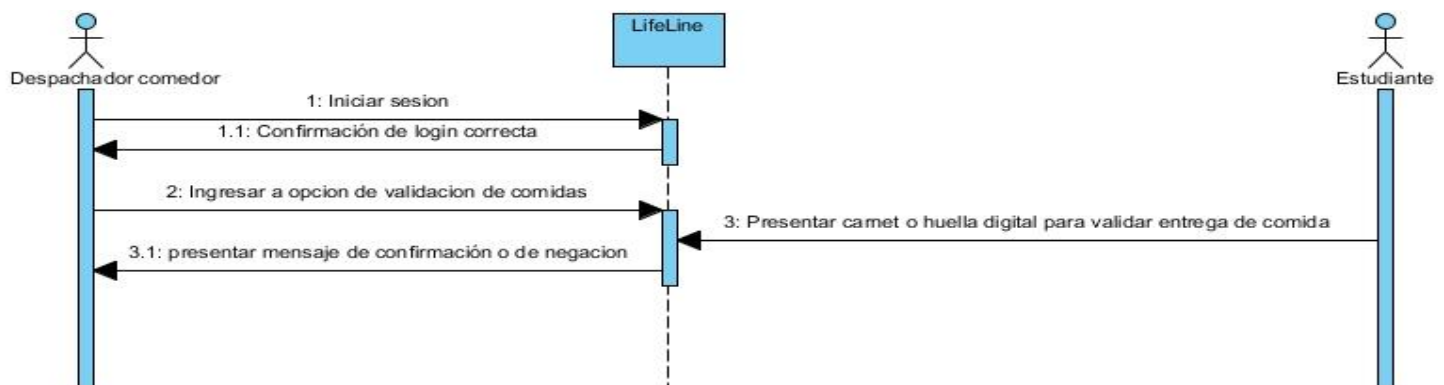


Diagrama 12: Proceso de validación de entrega de comida a estudiante

Diagrama de secuencia asociado a caso de uso: Asignación de actividades sociales por estudiantes, actor: Administrador

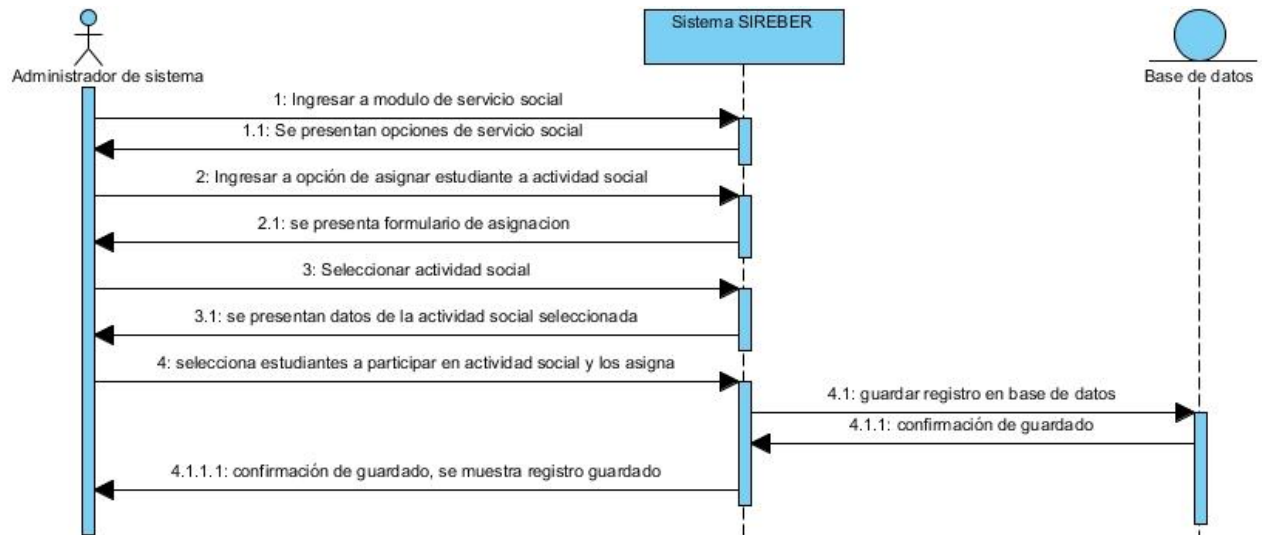


Diagrama 13: Proceso de asignación de actividad social por estudiante

Diagrama de secuencia asociado a caso de uso: Creación y activación de encuestas, actor: Administrador

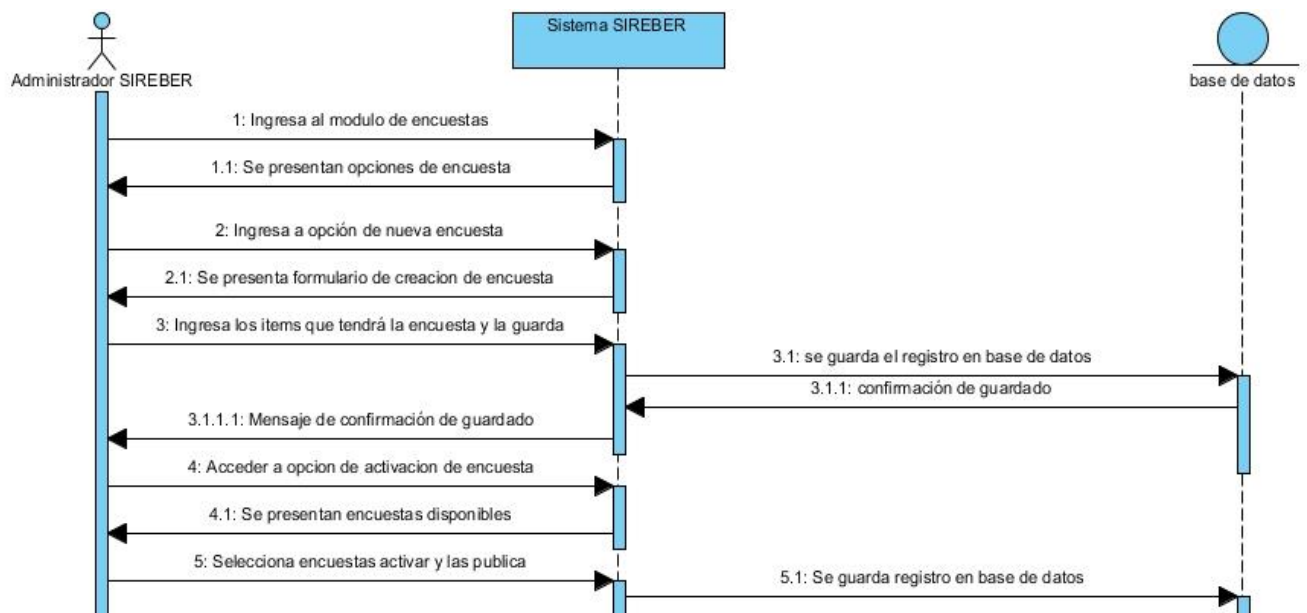


Diagrama 14: Proceso Creación y activación de encuestas

Diagrama de secuencia asociado a caso de uso: Contestar encuestas, actor: Estudiante.

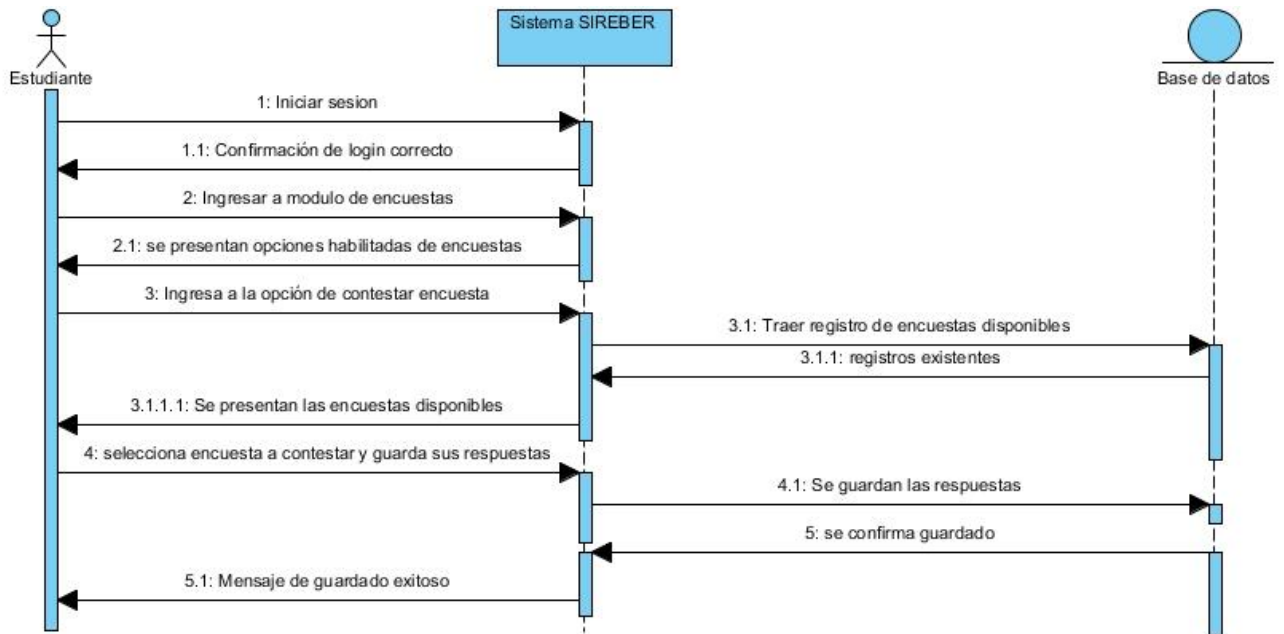


Diagrama 15: Proceso Contestar encuesta

6.2.1.5.7 Diagrama de contexto

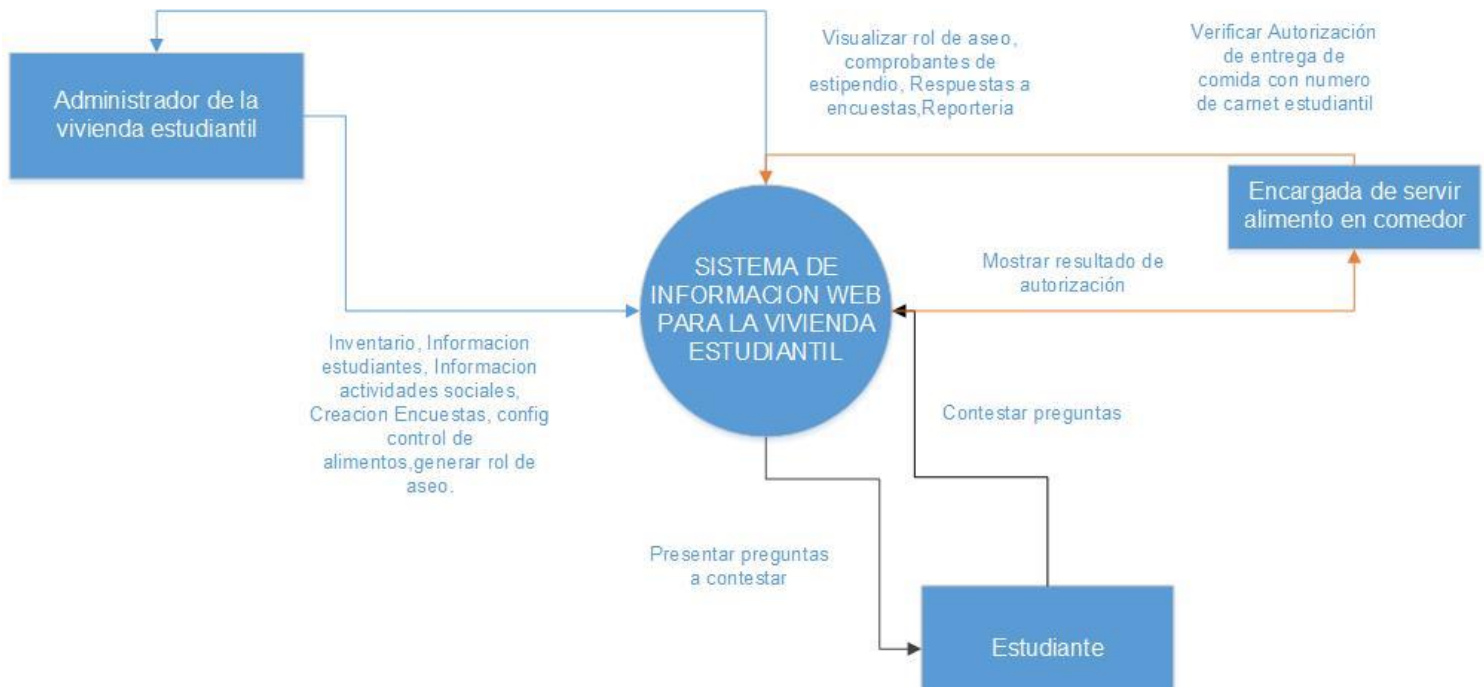


Diagrama 16: Contexto de sistema de información Web para la vivienda estudiantil

El lenguaje gráfico UML ayudó a visualizar, especificar, construir y documentar el sistema de información web para la gestión de la Vivienda Estudiantil. Esto permitió tener un respaldo de la lógica del sistema para poder entender cómo se manejan los procesos.

6.2.1.5.8 Codificación del sistema de información web para la gestión de la Vivienda Estudiantil y becados

Una vez terminado el diseño lógico, se realizó la codificación del sistema utilizando Visual Studio .NET, con MV5 y c#.

Como primer paso se creó una solución nueva, donde estarán todas las capas de programación con la que trabaja el sistema.

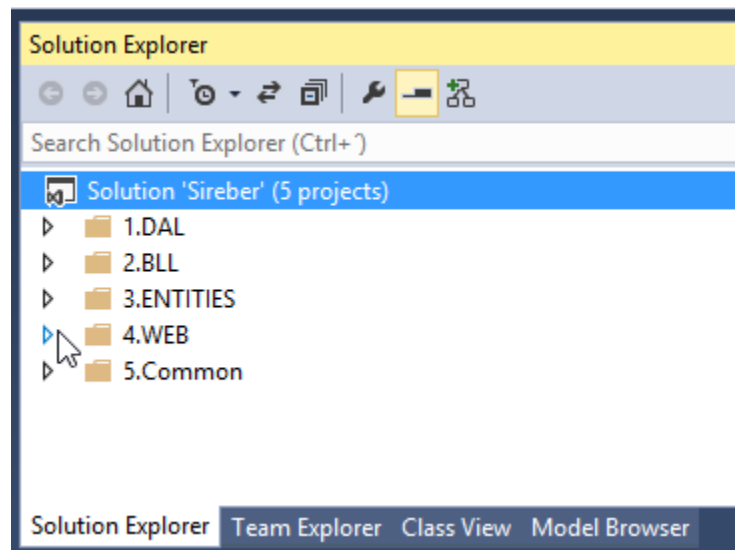


Figura 21: Solución en visual Studio del sistema de información web para la gestión de la vivienda estudiantil

Se codificó la capa de datos, donde se realizó un mapeo de la base de datos con la herramienta Entity Framework, el sistema genera el archivo de configuración el cual se usa como acceso a la base de datos.

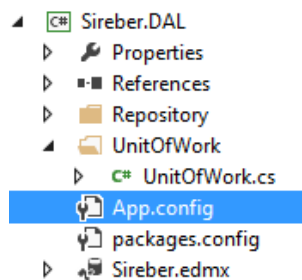


Figura 22: Archivos creados para acceso a datos por Entity Framework

En la siguiente imagen se presenta la cadena de conexión a la base de datos.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=2374
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFr
  </configSections>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />
    <providers>
      <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, Ent
    </providers>
  </entityFramework>
  <connectionStrings>
    <add name="SireberEntities" connectionString="metadata=res://*/Sireber.csdl|res://*/Sireber.ssdl|res://*/Sireber
  </configuration>
```

Figura 23: Contenido del archivo app.config

Una vez realizado el enlace a la base de datos, se procedió a codificar cada proceso descrito en la fase de diseño lógico para cada módulo del sistema de información web, para la gestión de Vivienda Estudiantil.

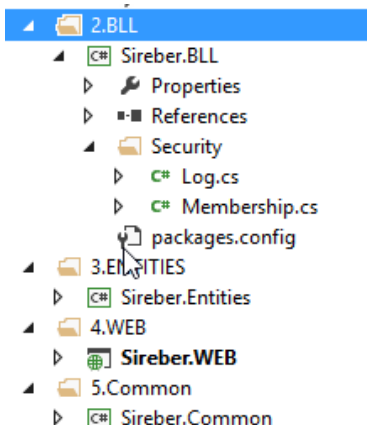


Figura 24: Codificación de los procesos administrativos de la vivienda estudiantil

En cada archivo que se muestra en la figura 24, están los métodos y funciones que conforman cada proceso administrativo.

6.2.2. Daily SCRUM

En esta etapa se hicieron reuniones de avance con el Scrum team, se plantearon algunas preguntas que guiaron el desarrollo de cada reunión, ¿Qué se hizo ayer?, ¿Qué se realizó hoy?. Se estudiaron los inconvenientes que surgían, y se superaban. Esta parte de los pasos desarrollados en la metodología SCRUM es de suma importancia, debido a que es donde se resolvió en equipo algunos inconvenientes, en cuanto a desarrollo de código o se estudiaban ciertos aspectos que no se comprendían.

6.2.3. Sprint Review

En esta etapa, ya se han finalizado cada sprint que se había planificado, y se presentaron avances funcionales del sistema al cliente. El cliente proporcionó retroalimentación desde su perspectiva sobre los adelantos mostrados y manifestó satisfacción con las funcionalidades entregadas en el sprint review de cada etapa.

7. Conclusiones y recomendaciones

7.1 Conclusiones

- Se logró desarrollar el sistema de información web para la gestión de la Vivienda Estudiantil y los servicios ofrecidos a los becados residentes de la UNI.
- El Scrum Team logró asimilar los procesos administrativos que se realizan en la gestión de la Vivienda Estudiantil y los becados residentes.
- El lenguaje UML complemento el proceso de asimilar los requerimientos.
- Se emplearon las técnicas normalización, patrón arquitectónico en capas, diseño adaptativo en el diseño de las interfaces, algoritmos y base de datos del sistema.
- El estudio de factibilidad que se realizó fue un punto importante ya dio la confirmación para empezar a trabajar en el desarrollo del sistema de información web para la gestión de la Vivienda Estudiantil y los becados residentes de la UNI.
- La metodología ágil de desarrollo SCRUM resultó ser muy completa y apropiada para el desarrollo del sistema, ya que permitió entregar partes funcionales al cliente y distribuir correctamente las tareas para cada requerimiento del usuario.
- El sistema de información web cumple con los requerimientos solicitados por el cliente, los cuales fueron detallados en las historias de usuario, como lo es ingresar inventario nuevo, configurar entrega de comidas, publicar encuestas para estudiantes y agendar actividad social.

7.2 Recomendaciones

- El sistema de información web es flexible y permite agregar nuevas opciones a los módulos existentes. Se recomienda agregar una opción en el módulo de Alimentación que permita hacer pedidos de comida a través del sistema.
- Implementar un módulo de inteligencia de negocios, el cual permitiría generar reportes dinámicos.
- Es indispensable que, las autoridades de la Dirección de Bienestar estudiantil, realicen las debidas gestiones con la Dirección de Tecnologías de Información y Comunicación, para la correcta implementación del sistema, así como, las capacitaciones al personal encargado de la manipulación del mismo.
- Realizar mantenimiento preventivo periódicamente en todos los equipos de cómputo, para garantizar una vida más longeva al servidor de base de datos y servidor web.
- Agregar funcionalidades para el envío de notificaciones y mensajes de textos a los estudiantes.
- Dar mantenimiento a la base de datos para verificar que todo esté bien a nivel de información y verificar espacio disponible en servidor de base de datos

8. Bibliografía

- Augusto, Enzo. (2011). *Administrador de servidores*. Mexico: Fox Andina.
- Booch, Grady. (2004). *El lenguaje unificado de modelado*. Mexico: Addison Welsey.
- Guthrie, Robert. (2015). *microsoft sql server*. Mexico: Pearson.
- Joyanes Aguilar, Luis. (2003). *Fundamentos de programación*. México: McGraw-Hill.
- Kendall, Kenneth. E., & E, K. Julie. (2005). *Análisis y diseño de sistemas* (Sexta ed.). México: Pearson.
- Laudon, Kenneth., & Laudon, Jane. (2012). *Sistemas de informacion gerencial*. México: Pearson.
- Menéndez, Rafael. (2011). *Bases de datos I*. México: Pearson.
- Mora, Lujan. (2001). *Programación en Internet Clientes Web*. Peru: Mesa redonda.
- O'Brien, J. A., & Marakas, G. M. (2006). *Sistemas de Información Gerencial*. México: McGraw Hill.
- Pressman, Roger. (2010). *Ingenieria del software "Un enfoque practico"* (Septima ed.). Mexico: McGraw-Hill.
- Puigvert, Antonio. (2004). *La Biblia de la Computacion e Internet*. Mexico: Lexus.
- Rob Pooley, Adison. W. (2002). *Utilización de UML en Ingeniería del Software*. México: Pearson.
- Sanchez, Jordan. (2006). *Usos de FrameWork*. Peru: Jordisan.
- Silberschatz, A., & Korth, H. F. (2002). *Fundamentos de base de datos*. España: McGraw-Hil.
- Sudarshan, Ali. (2011). *Fundamentos de base de datos*. Mexico: Pearson.
- Tanenbaum, Andrew. S. (2005). *Redes de computadoras* (Cuarta ed.). México: Pearson.
- Valdés, Damian. (2007). *Bases de datos*. Mexico: Pearson.
- Walls, Collin. (2005). *Embedded software*. México: Pearson.
- Wiley, John. (1997). *The Elements of User Interface Design* . Mexico: Mandel.

9. Glosario

- **Ataque CSRF (Del inglés Cross-site request forgery o falsificación de petición en sitios cruzados):**

Es un tipo de fragmento de código malicioso de un sitio web en el que comandos no autorizados son transmitidos por un usuario en el cual el sitio web confía. Esta vulnerabilidad es conocida también por otros nombres como XSRF, enlace hostil, ataque de un click, cabalgamiento de sesión, y ataque automático.

- **Certificado X.509:**

Es un estándar para infraestructuras de claves públicas (en inglés, Public Key Infraestructura o PKI). X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación

- **La nube de Internet (Cloud en inglés):**

Es un nuevo modelo de uso de los equipos informáticos. Traslada parte de tus archivos y programas a un conjunto de servidores a los que puedes acceder a través de Internet. Permite almacenar tus cosas en esos servidores. Y abrirlas, utilizarlas o usar programas que no están en tu equipo, sino en ellos.

Lo que normalmente estaría almacenado en tu PC (tus programas o tus archivos, por ejemplo) pasa a estar en los servidores que forman la nube. Por eso se habla en inglés de Cloud Computing, que suele abreviarse simplemente como The Cloud.

- **Load test:**

Las pruebas de carga es el proceso de poner la demanda en un sistema de software o dispositivo informático y medir su respuesta. Las pruebas de carga se realizan para determinar el comportamiento de un sistema, tanto en condiciones normales y esperadas de carga máxima.

- **OWASP (acrónimo de Open Web Application Security Project, en español ‘Proyecto abierto de seguridad de aplicaciones web’):**

Es un proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software sea inseguro.

- **Pruebas unitarias:** En programación, una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo en diseño estructurado o en diseño funcional una función o un procedimiento, en diseño orientado a objetos una clase. Esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado. Además de verificar que el código hace lo que tiene que hacer, verificamos que sea correcto el nombre, los nombres y tipos de los parámetros, el tipo de lo que se devuelve, que si el estado inicial es válido entonces el estado final es válido

- **Red privada virtual (RPV), en inglés: Virtual Private Network (VPN):**

Es una tecnología de red de computadoras que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet. Permite que la computadora en la red envíe y reciba datos sobre redes compartidas o públicas como si fuera una red privada con toda la funcionalidad, seguridad y políticas de gestión de una red privada. Esto se realiza estableciendo una conexión virtual punto a punto mediante el uso de conexiones dedicadas, cifrado o la combinación de ambos métodos.

- **Transport Layer Security (TLS, en español «seguridad de la capa de transporte»):**

Los certificados TLS son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

- **Web Performance Test:**

El grado en que un sistema o componente realiza sus funciones designadas dentro de las limitaciones dadas, tales como la velocidad, precisión, o el uso de la memoria. Durante estas pruebas no se busca encontrar errores, pero si se busca enfocarse en procesos individuales de la aplicación

10. Anexos

10.1 Anexo 1: Carta de solicitud de sistema web para la gestión de la Vivienda Estudiantil



10.2 Anexo 2: Estudio de factibilidad

El presente capítulo contiene el estudio de factibilidad del proyecto de implementación de un sistema de información web, para la gestión de la Vivienda Estudiantil y los becados residentes de la Universidad Nacional de Ingeniería.

Descripción de la situación actual

La Vivienda Estudiantil, como se ha descrito anteriormente, es un programa de la Universidad Nacional de Ingeniería (UNI), que a través de la Dirección de Bienestar Estudiantil (DBE), proporciona a los estudiantes que viven en los diferentes departamentos del país y tienen dificultad para viajar diario para asistir a la universidad, o no tienen los recursos financieros para alquilar un cuarto.

Los estudiantes de la UNI, residentes en la Vivienda Estudiantil, reciben los siguientes beneficios como parte del programa de becados internos, cuarto sin costo económico, alimentación, ayuda económica, enseres de aseo personal y de lavandería, muebles básicos para su dormitorio. Además, realizan y participan en actividades diarias que contribuyen a su formación integral, tales como: limpieza, charlas psicológicas, jornadas laborales y sobre todo su formación como ingenieros. Según el administrador de la vivienda, se atienden aproximadamente a unos ciento cincuenta estudiantes cada semestre.

Se hicieron reuniones periódicas con el administrador de la Vivienda Estudiantil. Se le entrevistó acerca de los problemas actuales que presenta, y comentó que la falta de herramientas y recursos tecnológicos que automaticen y agilicen la gestión de los procesos de administración de la Vivienda Estudiantil, ha causado

que los encargados, diariamente se den a la labor de llevar de forma manual, los procesos de administración y el control de las actividades de los becados residentes, desde registrar los datos personales de cada becado residente (y de ser necesario, actualizarlos), organizar y verificar el cumplimiento de los roles de aseo de las diferentes áreas de la vivienda, redactar llamados de atención en caso de que los estudiantes tengan mal comportamiento, llevar control de alimentación de los becados residentes, jornadas laborales, consultas médicas y psicológicas, hasta llevar el inventario de los inmuebles, control del laboratorio, los estipendios y ayuda de transporte que se le otorgan a cada universitario, durante el semestre.

Esta situación afecta incluso a los padres de los becados residentes, quienes se mantienen desinformados de las actividades que realizan sus hijos durante su estadía en la universidad, debido a que se carece de herramientas que puedan notificarles de las labores de sus hijos.

La problemática también perjudica a los becados residentes, al tener que esperar aproximadamente quince días para que se confirme su derecho a permanecer en la Vivienda Estudiantil, lo que los obliga a buscar hospedaje en instalaciones fuera de la universidad durante este período y a incurrir en gastos de hospedaje, alimentación, además de la inseguridad de los lugares.

Otro de los problemas es la entrega y recepción tardía de informes y/o reportes entre la Vivienda Estudiantil y las áreas que contribuyen en las diferentes actividades de los becados, áreas como las de tratamiento médico y psicológico, DBE, control de alimentación y horas laborales, se ven afectadas ante estos inconvenientes. Esto se debe a que, los informes son elaborados manualmente (en hojas de papel y luego en libros de Excel) y entregados de forma personal a cada uno de los encargados de las diferentes áreas, en consecuencia, se presentan demoras en las evaluaciones de cumplimiento de las metas

establecidas y en los tiempos de respuestas que se brindan ante las necesidades presentadas, en las que se ven mayormente afectados los becados residentes.

Por otra parte, se han presentado problemas referentes al control de la alimentación brindada a los becados residentes. Este control se lleva de forma manual, diariamente los estudiantes hacen filas para verificar si tienen derecho a retirar alimentos en el comedor de la vivienda, luego de ser verificados, deben presentar un ticket para recibir su orden de alimento, al final de cada horario de comida, las personas que atienden en el comedor deben entregar un reporte con la cantidad de órdenes de comida entregadas.

El problema radica no solo en la lentitud del proceso de retiro de alimentos, sino también en las inconsistencias de los reportes generados por los encargados del comedor, ya que generalmente las cantidades de servicios de comida registrados por los cocineros no corresponden a la cantidad de estudiantes que fueron enlistados para retirarlos, por consiguiente, los responsables del comedor deben asumir los costos de los servicios excedentes en los reportes, esto se debe a que no hay un proceso donde se registren que estudiantes efectuaron un retiro de alimentos en un mismo horario de comida.

Finalmente, la mayor preocupación de los responsables de la Vivienda Estudiantil y las autoridades de la DBE es que se continúe con estos problemas, pues de ser así, se seguirán registrando, entre otros: pérdidas económicas en las diferentes áreas, no será posible evaluar el cumplimiento y las metas establecidas, se incurrirá en mayores gastos de papelería y útiles de oficina, se invertirá mayor tiempo en procesos de gestión, se afectará a los estudiantes con los servicios y beneficios otorgados, se atrasarán las tomas de decisiones administrativas. No se podrán realizar auditorías a fin de corroborar el trabajo realizado en la Vivienda Estudiantil.

10.3 Estudio de requerimientos

Características de perfiles de usuarios

Tipo de perfil	Administrador del sistema
Actividades	-Atender las configuraciones de uso: Brindar soporte de primer nivel cuando se presenten inconvenientes en el funcionamiento normal del sistema, cabe destacar que el soporte es a nivel de configuraciones y no técnico. -Establecer perfiles de usuario: Puede crear perfiles de usuarios (por ejemplo, administrador de vivienda que tiene acceso a todas las opciones en el sistema) y establecer a que módulo o formulario tienen acceso cada usuario.

Tabla 3: Descripción de perfil administrador del sistema

Tipo de perfil	Administrador Vivienda Estudiantil
Actividades	-Tiene acceso a opciones administrativas del sistema: - Ingresar entradas y salidas de inventario. -Generar reportes varios, entre ellos reporte de gastos de estipendio, para visualizar información de cuánto se le ha asignado a cada estudiante en concepto de ayuda económica. Reporte de gastos de alimentación, que permite ver lo que se ha gastado en servicios de comida. Reporte de

	<p>movimientos en inventario, para artículos en existencia, etc.</p> <p>-Establecer configuración de horarios y turnos de comida, esto para dejar parametrizado cómo se va a controlar la entrega de comida en los comedores.</p> <p>-Ingresar a estudiantes nuevos en el sistema.</p> <p>-Ingresar información acerca de las actividades de servicio social de cada estudiante, para llenar tablas de histórico en el sistema.</p>
--	---

Tabla 4: Descripción de perfil administrador Vivienda Estudiantil

Tipo de perfil	Estudiantes
Actividades	<p>Opciones habilitadas:</p> <p>-Responder encuestas en el sistema. Una vez que se cree una encuesta, el estudiante recibirá una notificación, para que revise y conteste las encuestas publicadas.</p> <p>-Revisar asignaciones del día, una vez que el rol de aseo sea generado, el estudiante tiene un espacio para consultar la fecha y área le corresponde limpiar.</p>

Tabla 5: Descripción de perfil de estudiante



Figura 25: Jerarquía de perfiles de usuario (Elaboración propia)

En esta imagen se aprecia la jerarquía que tienen los perfiles de cada usuario. Como primer nivel se tiene al administrador del sistema, que es el usuario que tiene mayor control y conocimiento del sistema, ya que maneja opciones de configuración, control de roles, usuarios, control de menú, etc.

Posteriormente, está el perfil de administrador de la Vivienda Estudiantil, a diferencia del administrador del sistema, este solo tiene control y acceso sobre opciones administrativas, como control de inventario, alimentación y rol de aseo. El sistema está dirigido al área administrativa, por esta razón, dicho perfil cubre el 90% de las opciones disponibles.

El último perfil es el de estudiante, es el más limitado, ya que las opciones que tiene son formularios de consulta y el módulo de encuesta.

Cada usuario puede tener asignado más de un perfil.

Requisitos comunes de interfaces

Los requisitos comunes de interfaces se refieren a los requerimientos visuales que solicitó el cliente, por ejemplo, de qué colores quiere los formularios qué logo utilizar en la pantalla principal, etc.

Interfaces de usuario

Número de requisito	1
Nombre de requisito	Ventanas de visualización, menús y controles
Tipo	Requisito
Fuente del requisito	Administrador Vivienda Estudiantil
Prioridad del requisito	Alta
Detalle	La interfaz en general debe tener colores distintivos de la Universidad Nacional de Ingeniería, como son colores azules fuertes y tonos de azul bajos, combinados con blanco.

Tabla 6: Detalle de interfaces de usuario

Número de requisito	2
Nombre de requisito	Menú dinámico
Tipo	Requisito
Fuente del requisito	Administrador Vivienda Estudiantil
Prioridad del requisito	Alta
Detalle	La Interfaz debe tener lista de menús desplegables, dinámicas de acuerdo a roles de perfil de usuario.

Tabla 7: Detalle de interfaz de menú

Número de requisito	3
Nombre de requisito	Diseño responsive
Tipo	Requisito
Fuente del requisito	Administrador Vivienda Estudiantil
Prioridad del requisito	medio
Detalle	Se requiere una interfaz adaptable a cualquier tamaño de pantalla, usando un diseño responsive

Tabla 8: Descripción de diseño responsive

Interfaces de software

Las interfaces de usuario son necesarias cuando un sistema se debe comunicar con otro sistema externo, en este proyecto no se requiere un api (Interfaz de software) de comunicación para otro software ya que es un sistema propio de la Vivienda Estudiantil, no trabajará con otro sistema externamente.

10.4 Anexo 3: Factibilidad técnica

Para realizar el proyecto del sistema de información, la Vivienda Estudiantil cuenta con el apoyo de la División de Tecnologías de la Información y de Comunicación de la UNI (DTIC). Esta División pone a la disposición los recursos necesarios para llevar a cabo el proyecto.

Recursos de hardware existentes en la Universidad Nacional de Ingeniería

Tabla 1. Servidor	
Servidor DELL T320	
Cantidad	1
Procesador	Intel Xeon Cpu E5 2403V2 1.8Ghz Cuatro Núcleos
Memoria	RAM 4GB DDR3L
Disco Duro	Disco Duro 1TB 7200 RPM Serial ATA III
Otros	Teclado Mouse Monitor Escritorio Batería-Estabilizador 2 RJ45 y 1 Puerto Serial 2 Ranuras PCI Express x4
Fuente: Levantamiento de datos.	

Tabla 2. Computadora de desarrollo	
Toshiba Satellite L55-C5336S laptop	
Cantidad	2
Procesador	Intel Core I7-5500 2.4GHz
Memoria	RAM 8GB DDR3
Disco Duro	Disco Duro de 1TB
Otros	Teclado Mouse Monitor Escritorio Batería-Estabilizador 2 RJ45 y 1 Puerto Serial 2 Ranuras PCI Express x4
Fuente: Levantamiento de datos.	

Tabla 3. Computadoras de usuarios finales	
DELL	
Cantidad	5
Procesador	Dual core 2.4GHz
Memoria	RAM 4GB DDR3
Disco Duro	Disco Duro de 500 GB
Otros	Teclado Mouse Monitor Escritorio Batería-Estabilizador 2 RJ45 y 1 Puerto Serial 2 Ranuras PCI Express x4
Fuente: Levantamiento de datos.	

Accesorios de Computadoras existentes en la Universidad Nacional de Ingeniería

Accesorio	Cantidad
Batería UPS	3
Impresora	1

El equipo mencionado en las secciones anteriores, es suficiente para el desarrollo del sistema de información web para la Vivienda Estudiantil, se proporcionaron dos laptop con suficiente capacidad para trabajar y codificar el sistema, debido a que cuentan con un procesador con una rápida capacidad de respuesta y la velocidad necesaria para cumplir correctamente procesamientos de datos. La vida útil de buen desempeño para cada laptop es de 2 años.

Una vez que el proyecto se ponga en ejecución, se tiene a disposición un servidor web, donde se monta el sistema de información web para la Vivienda Estudiantil y la base de datos. El servidor cuenta con suficiente capacidad para soportar el sistema. La vida útil de un buen desempeño al servidor es de 2 años, siempre y cuando se le de un correcto mantenimiento.

Recursos de software existentes en la Universidad Nacional de Ingeniería

Para el proceso desarrollo del sistema, se presentan las siguientes propuestas de software, para las cuales se hará uso del hardware especificado anteriormente.

Tabla 4. Recursos de software existentes	
Sistema Operativo	Windows Server 2012, Windows 8
Gestor de Base de Datos	SQL SERVER 2014 R2
Lenguaje de Programación	ASP.NET MVC 5 C#
Servidor web	Internet information services (IIS)
Fuente: Levantamiento de datos.	

Tabla 5. Propuesta 1	
Sistema Operativo	Windows (incluido en equipo proporcionado)
Gestor de Base de Datos	SQL SERVER 2014 R2
Lenguaje de Programación	ASP.NET MVC 5
Herramientas Auxiliares	SQL SERVER 2014 (Diseño de Base de datos-Diagrama relacional), Visual paradigm (Diagramas UML para la Metodología de desarrollo seleccionada), Herramientas ofimáticas (Microsoft office 2013, incluidas en equipo proporcionado), Adobe Reader (Presentación de reportes del sistema).
Fuente: Análisis crítico en pareja.	

Tabla 6. Propuesta 2	
Sistema Operativo	Windows (incluido en equipo proporcionado)
Gestor de Base de Datos	MySQL
Lenguaje de Programación	Java (Netbeans)
Herramientas Auxiliares	MySQL Workbench (Diseño de Base de datos-Diagrama relacional), Power Designer Portable (Creación de Diagrama de Contexto), Rational Rose (Diagramas UML para la Metodología de desarrollo seleccionada), Herramientas ofimáticas (Microsoft office, incluidas en equipo proporcionado), Adobe Reader (Presentación de reportes del sistema).
Fuente: Análisis crítico en pareja.	

Comparación de propuestas

- Gestores de bases de datos

Característica	MySQL	SQL Server
Costo	Libre y de pago	De pago
Open Source	Sí	No
Plataformas	Linux, Windows y muchas otras	Sólo Windows
Límite de tamaño de la base de datos	Depende del almacenamiento del equipo	Depende del almacenamiento del equipo
Compatibilidad ACID	Depende del motor de almacenamiento	Sí
Transacciones	Sí	Sí
Servicio de reportes	No	Sí
Posibilidad de elegir diferentes formas de almacenamiento	Sí	No
Claves Foráneas	Depende del motor	Sí
Vistas	Sí	Sí
Procedimientos almacenados	Sí	Sí
Triggers	Sí	Sí
Cursorres	Sí	Sí
Subconsultas	Sí	Sí
Replicación	Sí	Sí
Funciones definidas por el usuario (UDF)	Sí	Sí

- Lenguajes de programación

Característica	ASP.NET	JAVA NETBEANS
Lenguajes Web	ASP MVC5	JSP
Capa de Control	COM/COM+	EJB (Enterprise JavaBeans)
Capa intermedia	ASP .NET	Servlets
Lenguajes Ap.	C#	Java
Web Services	Sí	Sí
Multiplataforma	Sí	Sí

Matriz de evaluación por ponderación			
Factores a Evaluar	Ponderador*	Propuesta 1	Propuesta 2
Ahorro de Costos Desarrollo-Operativos	10%	100	85
Tiempo de entrega	30%	100	95
Software utilizado	20%	90	80
Efectividad ³	40%	90	85
TOTAL (Total = \sum (Ponderador x Propuesta))	100%	94	87
Fuente: Elaboración propia.			

Cumple adecuadamente: 80-99
Cumple con restricciones: 60-79
Cumple con muchas restricciones: 40-59
No cumple: 0-39

**El ponderador es asignado por el grupo analizador, según los requerimientos planteados y entrevistas*

Ambas propuestas cumplen adecuadamente con los factores evaluados, sin embargo, la propuesta más atractiva es la Propuesta 1. Si bien es cierto, la propuesta 2, se hace uso del software libre, el cual no tiene costos de licenciamiento de software, sin embargo a largo plazo generaría gastos , ya que se necesitaría contratar personal especializado , debido a que no hay mucha información de ayuda del mismo, no se tiene garantía del producto adquirido ni seguridad. Por otro lado el software propietario, posee mayor soporte al momento de presentarse inconvenientes, hay actualizaciones periódicas de mejora, otra de las ventajas es la cantidad de ayuda en línea que existe del software propietario.

Al momento de invertir en software propietario, se garantiza la calidad del producto adquirido, además de la seguridad que proporciona la empresa propietaria del software, con todos los beneficios mencionados, se garantiza una buena inversión a largo plazo y definitiva.

³ Definición: Capacidad de lograr el efecto que se desea o se espera. DICCIONARIO DE LA LENGUA ESPAÑOLA - Vigésima segunda edición.
http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=efectividad

Debido a que el proyecto es un sistema de información web, el usuario final solo necesita un navegador, preferiblemente Google Chrome, debido a que este navegador es 100% compatible con la tecnología de diseño que se usa en el sistema web, otros navegadores como Firefox o Internet Explorer no trabajan adecuadamente con las plantillas CSS y HTML usadas en el proyecto y esto causaría una experiencia visual poco amigable al usuario.

Google Chrome cuenta con tecnología Webkit2, que es un motor de renderizado de HTML de código abierto, desarrollado por Apple a partir del código del proyecto KHTML, y es reconocido por ser sumamente liviano, tener una base de código prolija y limpia, cumplir estrictamente con los estándares y usar poca memoria para su funcionamiento. El Webkit2 sirve para poder generar la interfaz del sistema en el navegador.

Sistema de comunicaciones existentes en la Universidad Nacional de Ingeniería

La Universidad Nacional de Ingeniería (UNI) cuenta con servicio de Internet en las oficinas administrativas de la Vivienda Estudiantil y cobertura WiFi en las siguientes áreas:

- Edificio Rigoberto López Pérez
- Biblioteca “Esmán Marín”
- Edificio Facultad de Electrotecnia y Computación
- Oficinas administrativas Dirección de Bienestar Estudiantil

Este servicio de Internet garantiza el acceso al sistema de información web, de la Vivienda Estudiantil.

Recursos Humanos necesarios

El personal para el desarrollo del proyecto debe tener el siguiente perfil:

Cargo	Cantidad
Analista-Programador	2

Analista-programador:

Profesional analista, diseñador y programador de sistemas web. Excelente dominio en manejadores de bases de datos SQL Server 2014 y MySQL, así como el manejo de herramientas case y lenguajes de programación (Visual studio 2013 MVC5 y JAVA Netbeans). Deben ser capaces de dirigir con eficiencia y carácter proyectos informáticos. Capacidad para trabajar en grupo y bajo presión.

Conclusiones:

Según el análisis de los datos mostrados anteriormente, se concluye que el proyecto de desarrollo del sistema de información web para la gestión de la Vivienda Estudiantil y becados es factible técnicamente, ya que el equipo hardware con el que cuenta la Universidad Nacional de Ingeniería (UNI) es suficiente para desarrollar el sistema, así también el software existente en la UNI, cumple con las demandas del sistema.

10.5 Anexo 4: Factibilidad Operativa

El proyecto de sistema de información para la gestión de la Vivienda Estudiantil y los becados residentes de la Universidad Nacional de Ingeniería contará con la ayuda del personal de la División de Informática y Tecnologías de Información (DTIC), en cuanto a la implantación del sistema y los futuros mantenimientos, cabe destacar que tienen un personal con experiencia en estos tipos de proyectos y el personal es amplio para dar atención al sistema, una vez finalizada la implantación. Entre el personal con el que cuentan tienen analistas de sistemas, programadores y administrador de base de datos, una vez concluido el proyecto se capacitará al personal para que puedan realizar posteriormente, el mantenimiento.

El sistema de información web tendrá una interfaz sencilla e intuitiva, que permitirá al usuario un fácil uso del mismo, además el manejo no se limita a la computadora, también se podrá acceder y utilizar desde un teléfono inteligente o tableta. La ventaja de usar estos tipos de dispositivos, es que actualmente los usuarios están familiarizados con este tipo de equipos.

Como se detalló en la descripción del problema, se explica que los procesos de la administración de la Vivienda Estudiantil se llevan manualmente, y no se tiene una consistencia segura de los datos, este sistema ayuda a que esos procesos se agilicen y se guarde la información de manera confiable y segura.

El administrador de la Vivienda Estudiantil que será el principal usuario del sistema, se encuentra entusiasmado y dispuesto a usarlo y adaptarse al cambio, los estudiantes también están motivados para aprender a usarlo. Es importante

señalar que los jóvenes, están acostumbrados a manejar diversos sistemas y aplicaciones, lo que ayuda al momento de usar el sistema de la Vivienda Estudiantil.

Los Sistemas de Información han cambiado la forma en que operan las organizaciones actuales. A través de su uso se logran importantes mejoras; automatizan los procesos operativos y suministran una plataforma de información necesaria para la toma de decisiones.

Por estas razones se concluye que el sistema es operativamente viable, ya que la Universidad Nacional de Ingeniería cuenta con el personal calificado para manejarlo, además el mismo es intuitivo y se aprende a manejar de manera rápida.

10.6 Anexo 5: Factibilidad de cronograma

Para realizar la factibilidad de cronograma se usaron los puntos de función. Para determinar dicha información se analizó la información presentada en las plantillas de casos de uso (**Ver Anexo 11: Plantillas de caso de uso**).

Modulo	Inputs/Entradas externas	Outputs/Salida Externa	Files/Archivos lógicos internos	Interfaces/Archivos Externos de Interfase	Queries/Solicitudes externas
Inventario	3	0	2	0	2
Aseo	1	0	3	0	2
Reportes	1	1	7	0	2
Control estipendio	1	0	2	0	2
Servicio Social	3	0	2	0	2
Alimentación	4	0	4	1	2
Encuestas	2	0	2	0	2
Total por Proyecto:	15	1	22	1	14

Para realizar un cronograma estimado, se utilizó la herramienta COCOMO II. Según los datos arrojados por la herramienta, el proyecto se clasifica como intermedio, debido a la cantidad de tiempo y personal que se necesita. Para lograr estos resultados, se utilizaron métricas de puntos de función, que es un método utilizado en ingeniería del software para medir el tamaño del software y pretende medir la funcionalidad entregada al usuario independientemente de la tecnología utilizada para la construcción y explotación del software.

Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	11.2	7.9	369.0	4495.93	1.1	1.4	
Most Likely	14.0	8.5	295.2	5619.91	1.4	1.7	0.0
Pessimistic	17.6	9.1	236.1	7024.89	1.7	1.9	

Según la proyección que presenta la herramienta COCOMO II, el proyecto tiene una duración de 7.9 meses y se necesita solo 1.4 de personas para llevarlo a cabo, debido a que es un trabajo monográfico, equipo de dos estudiantes el tiempo estimado que se propuso es de 8 meses, para dejar así un tiempo disponible por cualquier inconveniente que se presente. Como se logra ver es un proyecto que puede realizarse en un período de tiempo razonable, y esto beneficia al administrador de la Vivienda Estudiantil, ya que está urgido por utilizar el sistema.

Cronograma de Actividades

A continuación se detalla el cronograma y la distribución de tiempo establecido para cada proceso en el desarrollo del proyecto.

		Task Mode ▾	Task Name ▾	Duration ▾	Start ▾	Finish ▾
0			▸ Cronograma de Actividades	200 days	Tue 1/3/16	Mon 5/12/16
1			▸ Definición del Proyecto	15 days	Tue 1/3/16	Mon 21/3/16
2			Definición del Alcance del y Objetivos del Proyecto	15 days	Tue 1/3/16	Mon 21/3/16
3			▸ Análisis de Requerimientos	16 days	Tue 22/3/16	Tue 12/4/16
4			Identificación de las Necesidades	5 days	Tue 22/3/16	Mon 28/3/16
5			Diseño Preliminar de Requerimientos	5 days	Tue 29/3/16	Mon 4/4/16
6			Definición de Requerimientos Funcionales / No Funcionales	5 days	Tue 5/4/16	Mon 11/4/16
7			Aprobación de Requerimientos	1 day	Tue 12/4/16	Tue 12/4/16
8			▸ Diseño	15 days	Tue 12/4/16	Mon 2/5/16
9			Diseños preliminares basados en los requerimientos	15 days	Tue 12/4/16	Mon 2/5/16
10			Diseño de Base de Datos	1 day	Tue 12/4/16	Tue 12/4/16
11			▸ Desarrollo del Sistema	90 days	Wed 13/4/16	Tue 16/8/16
12			Identificación y Desarrollo Inicial de los Módulos	10 days	Wed 13/4/16	Tue 26/4/16
13			Asignación de Módulos	3 days	Wed 27/4/16	Fri 29/4/16
14			Desarrollo Final de los Módulos	70 days	Mon 2/5/16	Fri 5/8/16
15			Pruebas Iniciales	7 days	Mon 8/8/16	Tue 16/8/16
16			▸ Pruebas	30 days	Wed 17/8/16	Tue 27/9/16
17			Pruebas Funcionales por Módulo	15 days	Wed 17/8/16	Tue 6/9/16
18			Pruebas de Integración entre Módulos	15 days	Wed 7/9/16	Tue 27/9/16
19			▸ Implementación del Sistema y Capacitación	17 days	Wed 28/9/16	Thu 20/10/16
20			Diseño de Materiales de Capacitación	5 days	Wed 28/9/16	Tue 4/10/16
21			Capacitaciones a Usuarios Administradores	5 days	Wed 5/10/16	Tue 11/10/16
22			Configuración de Equipos de Cómputo	7 days	Wed 12/10/16	Thu 20/10/16
23			Documentación	120 days	Tue 1/3/16	Mon 15/8/16
24			Redacción del Documento de Tesis	110 days	Tue 1/3/16	Mon 1/8/16
25			Mejoras	30 days	Mon 8/8/16	Fri 16/9/16
26			Presentación del Documento Final	1 day	Mon 19/9/16	Mon 19/9/16
27			Pre-Defensa	1 day	Wed 23/11/16	Wed 23/11/16
28			Defensa	1 day	Mon 5/12/16	Mon 5/12/16

Figura 26: Cronograma de proyecto Sistema de información web



Figura 27: Línea de tiempo de ejecución del proyecto

La **figura 27** muestra un resumen en tiempo, de lo consumido en el proyecto. Se puede apreciar cómo se distribuyen las tareas en cada mes.

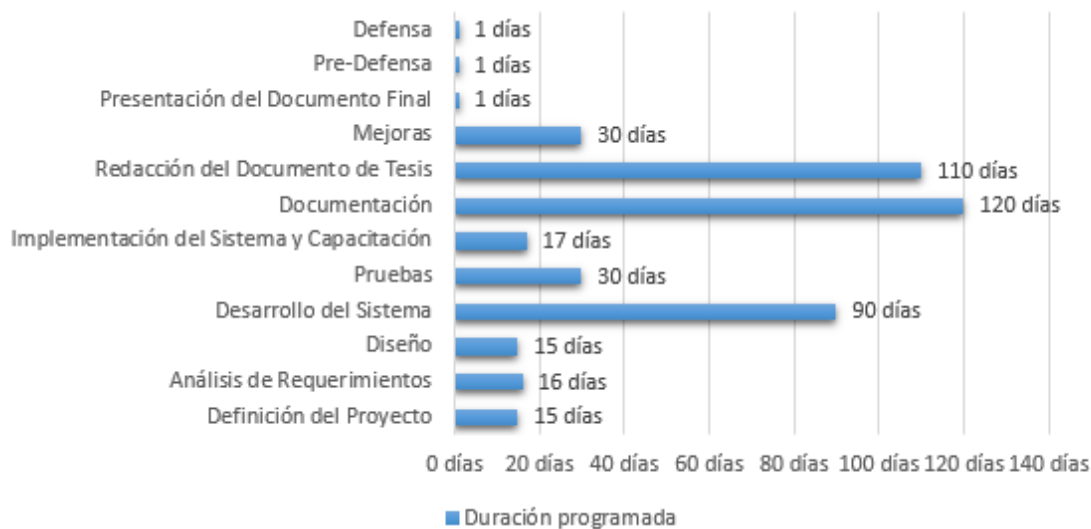


Figura 28 Resumen de Duración de Cronograma

En la **figura 28** se muestra un diagrama donde se pueden apreciar las tareas que demandan más días de ejecución.

Conclusión:

Según el estudio realizado con respecto al cronograma y tiempos de desarrollo del sistema de información web para la gestión de la Vivienda Estudiantil y los becados, es factible ya que el período de tiempo de entrega es razonable. Previendo cualquier inconveniente se estableció un margen de tiempo de un mes adicional.

10.7 Anexo 6: Factibilidad Legal

Accesos y permisos

La información no debe ser expuesta al público o a cualquier persona no autorizada, dentro o fuera de la Universidad Nacional de Ingeniería debido a que hay información del estudiante de índole privada y personal, por esta razón el administrador de la Vivienda Estudiantil solicitó que el sistema tenga gestión de accesos y permisos de usuarios y de esta manera controlar qué persona hace uso del sistema y controlar lo que puede ver dentro del sistema de información web.

También se consultó al encargado de la Oficina de Acceso a la Información Pública, el comentó que siempre se debe tener correctamente resguardada la información de cada estudiante, debido a que no puede ser usada por personas no autorizadas o que no tengan una relación con la institución que la use, también refirió que la Universidad Nacional de Ingeniería (UNI) hace uso de la ley de la información para normar en uso que se le da dentro de la UNI.

Licencias de software

La Universidad Nacional de Ingeniería (UNI) facilitará el software que se tiene en existencia (**Ver Anexo 3: Factibilidad técnica**). Esto ayudará a que no haya problemas con asuntos legales de licencia, respecto a programas fraudulentos o que estén manipulados. La UNI tiene licencia propietaria de los siguientes softwares

- SQL SERVER 2014 R2
- Visual Studio 2013
- Microsoft Office 2013

Derechos de uso:

El sistema de información web para la gestión de la Vivienda Estudiantil y becados es propiedad de la Universidad Nacional de Ingeniería a través de la Dirección de Bienestar Estudiantil.

Conclusión:

Luego de evaluar los aspectos anteriores se determinó que es factible y legalmente realizar el desarrollo del sistema de información web para la gestión de la Vivienda Estudiantil y becados ya que no se infringe en ninguna ley que pueda ocasionar algún inconveniente.

10.8 Anexo 7: Factibilidad económica

A continuación, se presenta el análisis de la factibilidad económica para el desarrollo del sistema web para la Vivienda Estudiantil. Donde se muestra información acerca del costo del equipo hardware y del software que se necesitan. Se lista también los recursos humanos necesarios y la inversión que se requiere realizar para pagar los honorarios de los desarrolladores.

Así también se señalan los beneficios de llevar a cabo el sistema de información web.

Costos de inversión de software

Tabla 9. Propuesta 1	
Sistema Operativo	Windows (incluido en equipo proporcionado)
Gestor de Base de Datos	SQL SERVER 2014 R2
Lenguaje de Programación	ASP.NET MVC 5 C#
Herramientas Auxiliares	SQL SERVER 2014 (Diseño de Base de datos-Diagrama relacional), Visual paradigm (Diagramas UML para la Metodología de desarrollo seleccionada), Herramientas ofimáticas (Microsoft office 2013, incluidas en equipo proporcionado), Adobe Reader (Presentación de reportes del sistema).
Fuente: Análisis crítico en pareja.	

Tabla 8. Propuesta 2	
Sistema Operativo	Windows (incluido en equipo proporcionado)
Gestor de Base de Datos	MySQL
Lenguaje de Programación	Java (Netbeans)
Herramientas Auxiliares	MySQL Workbench (Diseño de Base de datos-Diagrama relacional), Power Designer Portable (Creación de Diagrama de Contexto), Rational Rose (Diagramas UML para la Metodología de desarrollo seleccionada), Herramientas ofimáticas (Microsoft office, incluidas en equipo proporcionado), Adobe Reader (Presentación de reportes del sistema).
Fuente: Análisis crítico en pareja.	

En ambas propuestas de software, no se incurrirá en gastos para la adquisición del software establecido, ya que la Universidad Nacional de Ingeniería cuenta con los programas y sistemas especificados anteriormente.

Costos de inversión de Hardware de desarrollo

En el desarrollo del sistema de información web para la Vivienda Estudiantil, se utilizará el hardware presentado en la factibilidad técnica (**Ver Anexo 3: Factibilidad técnica**).

Todo este equipo es proporcionado por la Universidad Nacional de Ingeniería y por el administrador de la Vivienda Estudiantil, ya que se tiene en existencia hardware de desarrollo. No se incurrirá en gastos para adquisición de nuevos equipos de hardware.

Costos de desarrollo

Recursos Humanos:

Para el desarrollo del sistema se propone el siguiente personal:

Cargo	Cantidad
Analista-Programador	2

Según el cronograma que se realizó (**Anexo 5: Factibilidad de cronograma**), el análisis de los requerimientos tardará 16 días en concluir, durante este período, las personas a cargo del desarrollo, desempeñarán su función de analista de sistemas.

Durante la codificación y diseño del sistema web, serán los programadores del proyecto, sin obviar su tarea como analistas para la retroalimentación y mejora del sistema en desarrollo

Por ser un trabajo monográfico, los miembros del grupo no recibirán salarios durante el proceso de implementación del sistema, siendo esto factible y satisfactorio económicamente para la Dirección de Bienestar Estudiantil, que está a cargo de la vivienda para los becados.

Sin embargo, las autoridades encargadas deberán designar un administrador y encargados del mantenimiento del sistema para su implantación, siendo su responsabilidad la remuneración económica de éstos.

Beneficios de automatizar los procesos administrativos de la Vivienda Estudiantil

Automatizar es aplicar procedimientos automáticos a un aparato, proceso o sistema. En el caso del sistema de información web para la Vivienda Estudiantil, se automatizaron los procesos requeridos para la administración de la misma.

Beneficios tangibles

- Se evitarán pérdida en costos de comidas, ya que el sistema validará si un estudiante está autorizado para retirar comida.
- Se ahorra tiempo en la elaboración y procesamiento de encuestas, ya que en el sistema se publicarán y podrán ser contestadas por todo estudiante que accede al sistema.
- Los gastos en inventarios se reflejarán mejor, ya que se lleva un registro de los mismos.
- Los gastos en papelería se reducirán considerablemente, ya que varias gestiones que incurrían en gastos de papelería, ahora se registran y se ven en el sistema.
- Integridad de datos, se refiere a los valores reales que se almacenan y se utilizan en las estructuras de datos del sistema de información web, que ejercerá un control deliberado sobre todos los procesos que utilicen los datos para garantizar la corrección permanente de la información

Beneficios Intangibles

- Mejora la toma de decisiones al administrador de la vivienda, gracias a los reportes que estarán presentes en el sistema.

- Aumenta la transparencia e imagen organizativa de la Vivienda Estudiantil ya que los procesos administrativos se manejan con mayor control y la toma de decisiones se basan en datos respaldados por el sistema de información web para la Vivienda Estudiantil.
- Incrementará la satisfacción de trabajo de los usuarios principales, el administrador de la vivienda y los estudiantes.
- Optimizará tareas administrativas tediosas, por ejemplo redactar informe de inventario a mano, llevar control de gastos en inventario, controlar entregas de comida, elaborar y entregar invitaciones de eventos, llevar control de estudiantes activos, elaborar encuestas masivas, agendar eventos o actividades sociales e invitar vía e-mail a cada estudiante interno de la Vivienda Estudiantil.
- Mejora en los procesos de administración de la Vivienda Estudiantil
- Mayor exactitud en el procesamiento de datos

Cotización hardware de desarrollo

Producto	Cantidad	Proveedor consultado
Servidor DELL T20	1	SEVASA



Características

- Servidor Dell PowerEdge T20 Procesador Intel Xeon E3-1225 v3 3.2GHz
- 8M Cache
- Quad Core (84W). Memoria RAM 4GB (1x4G) 1600 MHz Con cuatro Slot de memoria Disco duro de 1TB 7.2K RPM SATA 3.5 Bahías internas de unidad de disco 2 10 Puertos USB
- 1 VGA
- 1 RJ45
- 2 Puertos PS2
- 1 Puerto COM y 2 Puertos Display Port
- Unidad DVD-RW 2 Ranuras PCI Express x16
- 1 PCI Express x1 y Puerto PCI Controladora de Red Intel 82579 Gigabit Ethernet 10/100/1000 Mbps

• Item # 5542 • Parte #

Precio:	\$ 539.00
Descuento:	\$ 60.00
IVA	\$ 71.85
Total:	\$ 550.85
Cantidad: <input type="text" value="1"/>	
 AGREGAR AL CARRITO	

Producto	Cantidad	Proveedor consultado
Toshiba Satellite L55	2	SEVASA



Características

- Toshiba Satellite L55D-C5204R Procesador AMD A8-7410 2.2GHz Disco Duro de 750GB Memoria Ram 6GB DDR3 Red Inalámbrica Cámara Web Bluetooth Tres Puertos USB Lector de Tarjetas Un Puerto Hdmi Un Puerto RJ45 Unidad de Dvd+/-Rw Teclado Numérico Integrado Pantalla de 15.6 Pulgadas Sistema Operativo Windows 8.1 Single

• Item # 5711 • Parte #

Precio:	\$ 669.00
Descuento:	\$ 154.00
IVA	\$ 77.25
Total:	\$ 592.25
Cantidad: <input type="text" value="1"/>	
 AGREGAR AL CARRITO	

Producto	Cantidad	Proveedor consultado
UPS Respaldo	3	SEVASA



Características

- UPS inteligente con pantalla LCD Capacidad 1000VA/600W Puerto USB para Datos 6 Toma Corriente con respaldo de Batería 2 Toma Corriente Exclusivos contra Sobretensiones Protección RJ-45
- RJ-11
- Fax y Modem Tiempo de Respaldo 30 Min Batería 12V 7Ah
- Item # 804 • Parte #

Precio:	\$ 92.50
IVA	\$ 13.88
Total:	\$ 106.38

Cantidad:

 **AGREGAR AL CARRITO**

Producto	Cantidad	Proveedor consultado
Impresora	1	SEVASA



Características

- Impresora Multifuncional Modelo 3635 Velocidad de impresión negro Hasta 8.5 ppm ISO y Hasta 20 ppm Borrador Velocidad de impresión Color Hasta 6 ppm ISO y Hasta 16 ppm Borrador Ciclo de trabajo (mensual)
- A4) Hasta 1000 Paginas Volumen de páginas mensual recomendado 100 a 200 Tamaños de soportes de impresión admitidos A4
- B5
- A6 y sobre DL Tipo de Papel común
- fotográficos
- Papel para Folletos
- Sobres y otros papeles especiales de inyección de tinta. Sin Bordes Cartucho HP 664 Negro y 664 Color Conectividad USB 2.0 Wifi Incluye Cable USB

Precio:	\$ 49.95
Descuento:	\$ 2.95
IVA	\$ 7.05
Total:	\$ 54.05

Cantidad:

 **AGREGAR AL CARRITO**

Cotización de software de desarrollo

Producto	Cantidad	Proveedor consultado	Precio
Licencia Windows server 2012	1	Microsoft	U\$560

Microsoft Windows Server 2012 R2 Standard OEM (2 CPU/2 VM)

by [Microsoft](#)

Platform : Windows 10, Windows 8, Windows 7, Windows XP

★★★★★ 47 customer reviews

Price: **\$560.00** & **FREE Shipping**. [Details](#)

Only 4 left in stock.

Want it tomorrow, July 20? Order within **4 hrs 50 mins** and choose **One-Day Shipping** at checkout. [Details](#)

Sold by [NVD TEK](#) and [Fulfilled by Amazon](#).

Edition: **R2 Base License (2CPUs/2VM)**

Additional License
(2CPUs/2VM)
from \$449.00

Base License (2CPUs/2VM)
\$299.00

**R2 Base License
(2CPUs/2VM)
\$560.00**

R2 Base License (4CPU/4VM)
\$1,340.84

- Full OEM Version of Windows Server 2012 R2 Standard Edition
- Simplified management console and clustering included
- Deduplication: Reduce wasted storage automaticall
- Virtualization: Run up to two virtual machines on up to two CPUs
- OEM Server License and Media; Server 2012 Standard requires CALS, which are not included with this product

Producto	Cantidad	Proveedor consultado	Precio
Licencia Windows 8.1	1	Microsft	U\$139.99

Microsoft Windows 8.1 - Full Version

by [Microsoft](#)

Platform : Windows 10, Windows 8

★★★★☆ ▾ 1,466 customer reviews

Available from these sellers.

- This operating system is eligible for a free upgrade to Windows 10 when available. More details below.
- Get to it all from the new Start screen, even your familiar desktop
- Personalize with more tile sizes, colors, & backgrounds. Do more with side by side views of apps and sites
- Access photos & files virtually anywhere with OneDrive built-in
- Search, browse, and share more securely and quickly

5 new from \$139.99

Producto	Cantidad	Proveedor consultado	Precio
Licencia Visual Studio 2013 Ultimate	1	Microsoft	U\$190

Visual Studio 2013 Ultimate

by [MS](#)

★★★★☆ ▾ 1 customer review

Available from these sellers.

- <http://www.componentsource.com/products/microsoft-visual-studio-premium/index.html>

1 open box from \$190.00

Producto	Cantidad	Proveedor consultado	Precio
Licencia SQL SERVER 2014	1	Microsoft	U\$2584

SQL Server Standard Edition 2014 English US Only DVD 10 Clt

by [Microsoft](#)

Platform : Windows 2000

★★★★★ ▾ [1 customer review](#)

Price: **\$2,584.99** & FREE Shipping. [Details](#)

Producto	Cantidad	Proveedor consultado	Precio
Microsoft Office 2013	1	Microsoft	U\$124

Microsoft Office 2013 - Full Professional Version Product Key Card for 1 User (32&64 Bit)

by [Microsoft Office Suites](#)

Platform : Windows 7

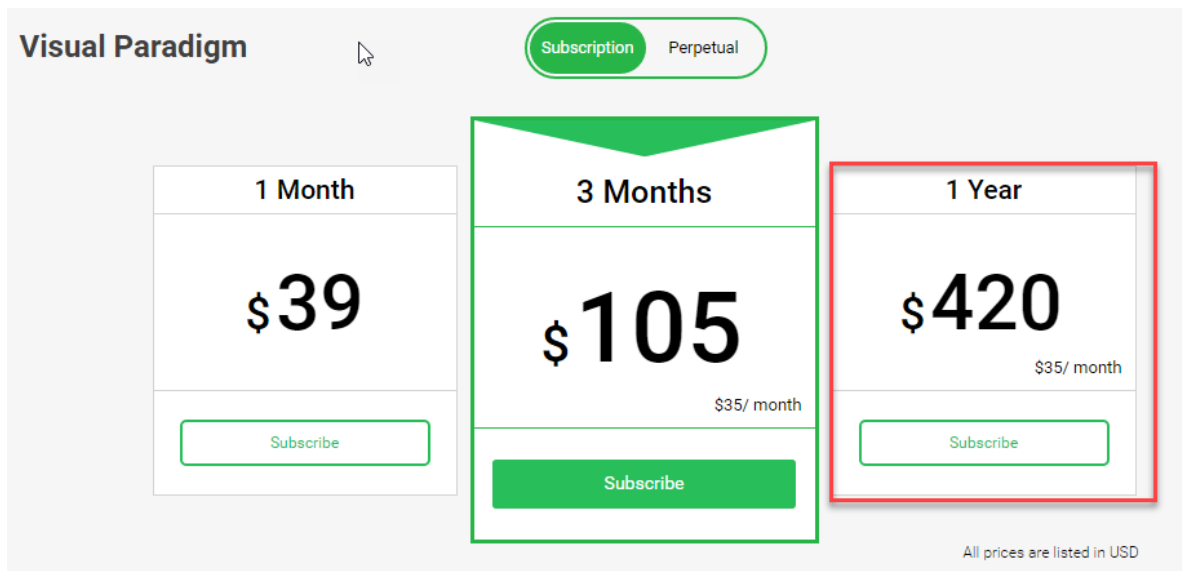
★★★★☆ ▾ [18 customer reviews](#)

[Available from these sellers.](#)

- DVD + KEY CARD

[21 new from \\$124.00](#)

Producto	Cantidad	Proveedor consultado	Precio
Licencia Visual Paradigm	1	Visual Paradigm Shop Online	U\$420 por suscripción de un año



Costos de recursos Humanos:

Personal	Cantidad	Salario mensual
Analista programador	2	U\$400

Según el cronograma (**Ver Anexo 5: Factibilidad de cronograma**) el proyecto tiene una duración de 8 meses, por cualquier inconveniente se preparó un pago de 9 meses, en pago de salario a los analistas programadores a cargo del proyecto, la inversión total en pago de recursos humanos es: $(\$400 \times 9 \text{ meses}) \times 2$, dando un total de \$7200 para pagar a dos programadores senior.

Costo estimado del proyecto

Para ambas propuestas de desarrollo (Software), la inversión estimada es, ver detalle de cotizaciones (**Ver Anexo 8: Cotización hardware de desarrollo**):

Tabla 9. Costos estimados del proyecto					
Costos de inversión		Costos de desarrollo		Costos complementarios	
Software	\$ 4017.99	RRHH	\$ 7200	Materiales Consumibles	\$ 0.00
Hardware	\$ 2108.54				
TOTAL DE PROPUESTA					\$ 13326.53
Fuente: Elaboración propia.					

La inversión total en el proyecto es de **\$13326.53 dólares**, este costo se lo está ahorrando la Universidad Nacional de Ingeniería ya que cuenta con el equipo que se debe comprar, así como también el software de desarrollo.

Conclusión:

Según los datos que se han mostrado, se concluye que el desarrollo del sistema de información web para la gestión de la Vivienda Estudiantil y becados es convenientemente factible económicamente ya que se consideran recursos existentes en la Universidad Nacional de Ingeniería como es el equipo hardware y software que se mencionó. Los recursos humanos no generaran gastos ya que es un trabajo monográfico y el equipo de desarrollo no cobrará por el trabajo realizado.

10.9 Anexo 8: Entrevista de levantado de requerimientos

La encuesta es una serie de preguntas que se hace a una o varias personas para reunir datos o para detectar la opinión pública sobre un asunto determinado.

Se realizó la siguiente encuesta al Licenciado Humberto Pérez administrador de la Vivienda Estudiantil, para conocer los procesos administrativos que el lleva a cabo y saber lo que el necesita que haga el sistema de información web.

Vivienda estudiantil procesos

Encuesta para conocer la opinión del encargado de la vivienda estudiantil a cerca de los procesos actuales de la vivienda estudiantil y en que aspectos se puede mejorar.

*Obligatorio

1. Que piensas que se debe mejorar en la vivienda estudiantil *

2. Que procesos son mas tardados o complicados para usted? *

3. Usted de qué se encarga en la vivienda estudiantil? *

4. Entrega informes a otras áreas a cerca de las gestiones que realiza en la vivienda estudiantil?

5. Tiene acceso a Internet y computadora? *

Marca solo un óvalo.

☐ Si
☐ No

https://docs.google.com/forms/d/_DQrYnD5Q5SkVAJF55Yla5YT4n7PlwL8ov3uMaledi

1/2

09/09/2016

Vivienda estudiantil procesos

6. Describa que realiza en cada proceso administrativo *

10.10 Anexo 9: Documento de proceso de observación

Acta de Verificación de Observación Presencial

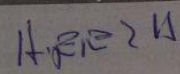
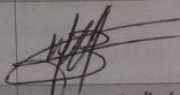
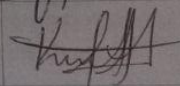
Proyecto: Desarrollo de sistema de información web para la vivienda estudiantil.

Fecha Elaboración: Sábado 5 de Marzo de 2016 y sábado 12 de marzo de 2016.

Lugar: Universidad Nacional de Ingeniería, Recinto Universitario Simón Bolívar, Vivienda Estudiantil

Duración: dos sesiones de 3 horas.

Participantes:

Nombres y apellidos	Función en el proyecto	Firma Aprobado
Humberto Pérez	Responsable de vivienda estudiantil y encargado del seguimiento del proyecto.	
Francel Martínez	Analista-programador	
Kenneth Gaitán	Analista-Programador	

Lista de chequeo procedimientos administrativos

11.10.21

Proceso	Pasos
Control de Cuartos Inventario	<ul style="list-style-type: none"> - Asignación de cuartos por estudiante - Registro de nuevo artículos de inventario. - Asignación de artículos por estudiante y cuarto. - Registro de movimientos de inventarios (Entradas, salidas, asignaciones). - Reporte de existencias y movimientos de inventario. - Visualización de inventario actual
Rol de aseo	<ul style="list-style-type: none"> - Generar rol de ase por semestre, semana, mes. - El rol generado debe incluir las áreas principales de la vivienda estudiantil. - No se deben repetir alumnos asignados a cada rol. - Generar rol de aseo de semestre por cuarto y para cada estudiante.
Reportes	<ul style="list-style-type: none"> - Existencias de inventario. - Detalle rol de aseo. - Encuestas con preguntas contestadas - Detalle de pedidos de comida - Actividades sociales realizadas - Comprobantes estipendio
Encuestas	<ul style="list-style-type: none"> - Registrar encuesta - Agregar, editar y eliminar respuestas de las encuestas - Compartir enlace de encuesta - Generar reportes analíticos basados en las respuestas de la encuesta
Servicio social	<ul style="list-style-type: none"> - Registrar Actividades con su respectiva fecha de inicio y fin, lugar de realización, hora de inicio y fin y descripción de la actividad. - Enviar invitaciones de actividades por medio de Email. - Organizar grupos. - Asignar funciones en servicio social. - Asignar porcentaje de cumplimiento de estudiantes a las actividades a las que fueron invitados.

11.12.19

Alimentación	<ul style="list-style-type: none">- Mantenimiento de horarios de alimentación- Mantenimientos de costos de servicios alimenticios- Control de retiro de alimentos en comedores- Mantenimiento de permisos de retiro de alimentos en recintos foráneos.- Autorización manual de retiro de alimentos- Generar Reportes de estudiantes a participar.- Generar Reporte de servicios alimenticios retirados por día
Memorándum	<ul style="list-style-type: none">- Registro de Memorándums por estudiantes
Catálogos	<ul style="list-style-type: none">- Mantenimiento de catálogos de sistema (Recintos, viviendas, carreras, semestres, marcas, modelos y días feriados)

10.11 Anexo 10: Lista de ISO, Usadas principales

- ISO 25010: 2011

El objetivo general de la creación del estándar es organizar, enriquecer y unificar las series que cubren dos procesos principales: especificación de requisitos de calidad del software y evaluación de la calidad del software, soportada por el proceso de medición de calidad del software.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura:



- ISO 9126

Es un estándar internacional para la evaluación de la calidad del software. El estándar está dividido en cuatro partes las cuales dirige realidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

Funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad y calidad de uso.

- **ISO 12207**

Es un estándar internacional para los procesos del ciclo de vida del software. Su objetivo es ser el estándar que define todas las tareas necesarias para el desarrollo y mantenimiento de software.

La norma tiene el objetivo principal de proporcionar una estructura común para que los compradores, proveedores, desarrolladores, mantenedores, operadores, gestores y técnicos que trabajan en el desarrollo de software utilicen un lenguaje común, que se establece en forma de procesos bien definidos.

La estructura de la norma estaba destinado a ser concebido de una manera flexible, modular de manera que sea adaptable a las necesidades de quien lo usa. La norma se basa en dos principios básicos: la modularidad y la responsabilidad. La modularidad significa procesos con un mínimo de acoplamiento y la máxima cohesión. Responsabilidad indica establecer una responsabilidad para cada proceso, lo que facilita la aplicación de la norma en proyectos en los que muchas personas pueden participar legalmente.

- **ISO 23270: 2006**

Especifica la forma y establece la interpretación de los programas escritos en el lenguaje de programación C #.

- **ISO 9241**

Es un estándar de usabilidad. Se usa para diseñar una estación de trabajo, evaluar una pantalla, evaluar una interfaz gráfica de usuario, probar un nuevo teclado, evaluar un nuevo dispositivo de interacción como una palanca de mando, comprobar que el entorno de trabajo está a la altura, y medir los reflejos y colores en una pantalla de visualización. Contiene listas de comprobación para ayudar a estructurar una evaluación de la usabilidad.

10.12 Anexo 11: Plantillas de caso de uso

Módulo de inventario:

Tabla 10. Casos de uso No. 1.	
Nombre	Registro de enseres / inventario nuevo en bodega
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Iniciar sesión en el sistema SIREBER. Se debe tener una lista actualizada y preparada anteriormente con los nuevos artículos a almacenar en bodega. Los artículos deben estar clasificados correctamente en la base de datos.
Flujo Normal	El administrador redacta la lista, se dirige al formulario de ingreso nuevo y los ingresa uno a uno, hasta actualizar toda la información en el sistema.
Flujo Alternativo	El administrador puede preparar un Excel con todos los artículos y subirlos masivamente, a través de la opción de subir masivo en el sistema.
Pos condiciones	La información se carga y se actualiza directamente en la base de datos.
Fuente: Elaboración propia.	

Tabla 11. Casos de uso No. 2.	
Nombre	Asignación de artículos por estudiantes y por cuarto
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Iniciar sesión en el sistema SIREBER. Se debe comprobar en el sistema, si hay artículos en existencia que se puedan consumir. Confirmar si realmente se necesita el reemplazo del artículo.
Flujo Normal	El administrador va al formulario de asignación de artículos, busca al estudiante o cuarto, selecciona el artículo a asignar, y da guardar.
Flujo Alternativo	Ninguna.

Pos condiciones	La información de asignación se guarda en la base de datos, y se actualiza la información de cada artículo en la tabla de artículos de inventarios. Internamente también se guarda una bitácora de gastos por estudiantes.
Fuente: Elaboración propia.	

Tabla 12. Casos de uso No. 3.	
Nombre	Baja de artículos por cuarto y estudiantes
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Iniciar sesión en el sistema SIREBER. Tener lista detallada de las bajas que hubo de artículos.
Flujo Normal	El administrador busca el formulario de bajas de artículos, busca y selecciona al cuarto o estudiante y cuando lo selecciona el sistema muestra los artículos existentes, el administrador selecciona los artículos a dar de baja y les da guardar.
Flujo Alternativo	Ninguna.
Pos condiciones	La información de asignación se guarda en la base de datos, y se actualiza la información de cada estudiante o cuarto.
Fuente: Elaboración propia.	

Tabla 13. Casos de uso No. 4.	
Nombre	Reporte de existencias y gastos generales
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Iniciar sesión en el sistema SIREBER. Debe existir información actualizada en la base de datos para que el reporte de una información confiable.
Flujo Normal	El administrador accede la pantalla de reportes, busca el reporte de información de existencias y gastos. Lo genera y exporta el reporte o lo visualiza en el sistema
Flujo Alternativo	Ninguna.
Pos condiciones	La información es mostrada.
Fuente: Elaboración propia.	

Módulo de alimentación:

Tabla 14. Casos de uso No. 5.	
Nombre	Registro de Recintos universitarios y comedores disponibles
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Tener una lista de recintos universitarios o comedores aprobados a funcionar en el sistema. Iniciar sesión en el sistema SIREBER.
Flujo Normal	El administrador se dirige al módulo de alimentación, busca la pantalla de catálogo de recintos universitario, digita la información correspondiente y le da guardar. Así mismo para los comedores disponibles el administrador se va a la pantalla de catálogos de comedores, digita la información nueva y da click en guardar. Posteriormente se ligan los estudiantes a ese comedor
Flujo Alternativo	Ninguna.

Tabla 15. Casos de uso No. 6.	
Nombre	Configurar entrega de comidas a estudiantes mediante carnet en los comedores
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Tener los parámetros de configuración de los comedores con respecto a los estudiantes correctamente. Es decir, los estudiantes deben estar asignados correctamente a los comedores existentes. Los carnet deben estar correctamente asignados. El administrador debe inicia sesión en el sistema SIREBER.
Flujo Normal	El administrador debe ir al módulo de alimentación, buscar la pantalla de validación de alimentos, selecciona comedor a configurar y define los parámetros de entrega alimento, como lo son hora de entrega, fecha, comedor asignado, tipo de comida y validar identificación de estudiante. Una vez validada la información el administrador le da guardar
Flujo Alternativo	Si el validador biométrico no está disponible, el administrador sigue el mismo proceso, con la diferencia de que el administrador habilita la validación de entrega por medio del carnet de estudiante
Pos condiciones	Una vez que el estudiante presente su carnet, el sistema valida si tiene comida disponible, el encargado del comedor actualiza la entrega en el sistema, y el estudiante ya no tiene derecho a otro retiro, hasta que se renueve otra entrega. El sistema internamente guarda un histórico de comidas entregadas por estudiante y sus costos.
Fuente: Elaboración propia.	

Tabla 16. Casos de uso No. 7.	
Nombre	Cargar menú de comida y su costos
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Se le debe pasar al administrador del sistema un listado del menú y el costo por plato. Debe iniciar sesión en el sistema SIREBER
Flujo Normal	El administrador se dirige al catálogo de menú. Ingresa las opciones de comida, selecciona el comedor, y los costos por plato, posteriormente le da guardar.
Flujo Alternativo	Se llena un Excel del menú en una plantilla proporcionada por el sistema, y lo carga masivamente en la misma pantalla de catálogo de menú
Pos condiciones	Ninguna
Fuente: Elaboración propia.	

Tabla 17. Casos de uso No. 6.	
Nombre	Validar entrega de comidas a estudiantes, mediante Numero de Carnet estudiantil
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Despachador comedor, estudiante
Precondiciones	Los parámetros para controlar la entrega de comida deben están configurados. El usuario debe acceder al sistema
Flujo Normal	El despachador verifica en el sistema cuando el estudiante se identifique, que el sistema acepte la entrega de comida
Flujo Alternativo	El despachador solicita número de carnet a estudiante para validar si está permitido y vigente entrega de comida.

Pos condiciones	Una vez que el estudiante presente su carnet, el sistema valida si tiene comida disponible, el encargado del comedor actualiza la entrega en el sistema, y el estudiante ya no tiene derecho a otro retiro, hasta que se renueve otra entrega de comida. El sistema internamente guarda un histórico de comidas entregadas por estudiante y sus costos.
Fuente: Elaboración propia.	

Módulo de servicio social:

Tabla 18.Casos de uso No. 9.	
Nombre	Crear actividades de servicio social
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Se debe tener la información correctamente revisada y preparada. El administrador debe iniciar sesión en el sistema.
Flujo Normal	El administrador accede a la pantalla para crear eventos de actividades sociales, ingresa las fechas y duración de la actividad, y posteriormente la activa y la guarda
Flujo Alternativo	Ninguna.
Pos condiciones	La actividad social queda activa para una futura asignación de participantes. Posteriormente el administrador puede desactivarla o modificar los datos.
Fuente: Elaboración propia.	

Tabla 19.Casos de uso No. 10.	
Nombre	Asignación de actividades sociales por estudiantes.
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador

Precondiciones	Tener una lista de los estudiantes a asignar y el evento debe estar creado y activo. El administrador debe iniciar sesión en el sistema
Flujo Normal	El administrador accede a la pantalla de asignación de actividades sociales, se le presenta los estudiantes existentes en la base de datos, los selecciona, luego selecciona la actividad social, los asigna y le da guardar.
Flujo Alternativo	Ninguna.
Pos condiciones	La información se guarda. Y el administrador puede visualizar quiénes son los que están asignados a esta actividad social. Y toda la información correspondiente dicha actividad. Esta información se puede modificar por el administrador
Fuente: Elaboración propia.	

Tabla 20. Casos de uso No. 11.	
Nombre	Ingreso de horas de trabajo voluntario por estudiante
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	El administrador debe registrar en una lista las horas de trabajo voluntario por estudiante. El administrador debe acceder al sistema e iniciar sesión.
Flujo Normal	El administrador se dirige al módulo de servicio social, se va a la pantalla de cargar horas de trabajo voluntario a estudiantes, ingresa la información correspondiente y la guarda.
Flujo Alternativo	Ninguna.
Pos condiciones	La información se guarda. Estará disponible para reportes de información.
Fuente: Elaboración propia.	

Módulo de encuestas:

Tabla 21. Casos de uso No. 12.	
Nombre	Creación y activación de encuestas
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	El administrador debe iniciar sesión.
Flujo Normal	El administrador accede al módulo de encuestas, va a la pantalla de creación de encuesta y le da en la opción de crear una, ingresa los ítems o pregunta que tendrá esa encuesta, la activa y le da guardar.
Flujo Alternativo	Ninguna.
Pos condiciones	La información se guarda y la encuesta queda publicada para que los estudiantes contesten.
Fuente: Elaboración propia.	

Tabla 22. Casos de uso No. 13.	
Nombre	Contestar encuestas
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Estudiante
Precondiciones	El estudiante debe iniciar sesión. Debe existir encuestas activas.
Flujo Normal	El estudiante accede al módulo de encuestas, selecciona las encuestas disponibles y las contesta, luego guarda los resultados.
Flujo Alternativo	Ninguna.
Pos condiciones	La información se guarda.
Fuente: Elaboración propia.	

Tabla 23. Casos de uso No. 14.	
Nombre	Visualización de resultados y encuestas en general
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	Deben existir encuestas contestadas. El administrador debe iniciar sesión en el sistema
Flujo Normal	El administrador ingresa al módulo de encuestas, se va a la parte de visualizar resultados. Selecciona la encuesta a consultar y da click en visualizar.
Flujo Alternativo	Ninguna.
Pos condiciones	La información es mostrada. No se puede modificar, solo esta para su lectura.
Fuente: Elaboración propia.	

Módulo de Reportes:

Tabla 24. Casos de uso No. 14.	
Nombre	Visualización de Información
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	La información debe estar guardada correctamente en las tablas de histórico. El administrador debe iniciar sesión.
Flujo Normal	El administrador ingresa al módulo de reportes, selecciona el reporte a visualizar, ingresa los parametros, y da click en visualizar.
Flujo Alternativo	Ninguna.
Pos condiciones	La información es mostrada.
Fuente: Elaboración propia.	

Módulo de Aseo:

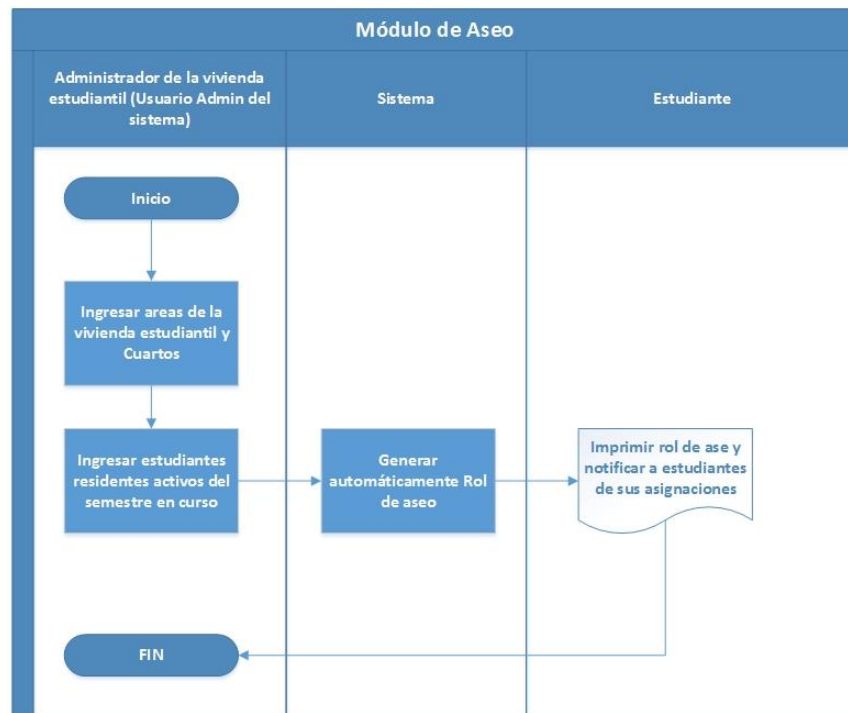
Tabla 25. Casos de uso No. 14.	
Nombre	Generar Rol de aseo
Autor	Grupo Monográfico
Fecha	12/04/2016
Actores	Administrador
Precondiciones	La tabla de áreas debe estar actualizada, al igual que la tabla de estudiante.
Flujo Normal	El administrador ingresa al módulo de Aseo, ingresa el rango de fechas correspondiente al rol de aseo, y da generar
Flujo Alternativo	Ninguna.
Pos condiciones	La información es mostrada. El sistema genera automáticamente el rol de aseo, y asigna a cada estudiante a ciertas áreas o cuartos.
Fuente: Elaboración propia.	

Con las plantillas de caso de uso se analizó el flujo y la interacción que tiene el administrador de la Vivienda Estudiantil con los procesos que el lleva a cabo en la misma, esto ayuda a entender mejor cada requerimiento. Posteriormente las plantillas de caso de uso se usaron como base para elaborar los diagramas UML (Diagrama de casos de uso, Diagrama de actividades, diagrama de secuencia, diagrama de contexto).

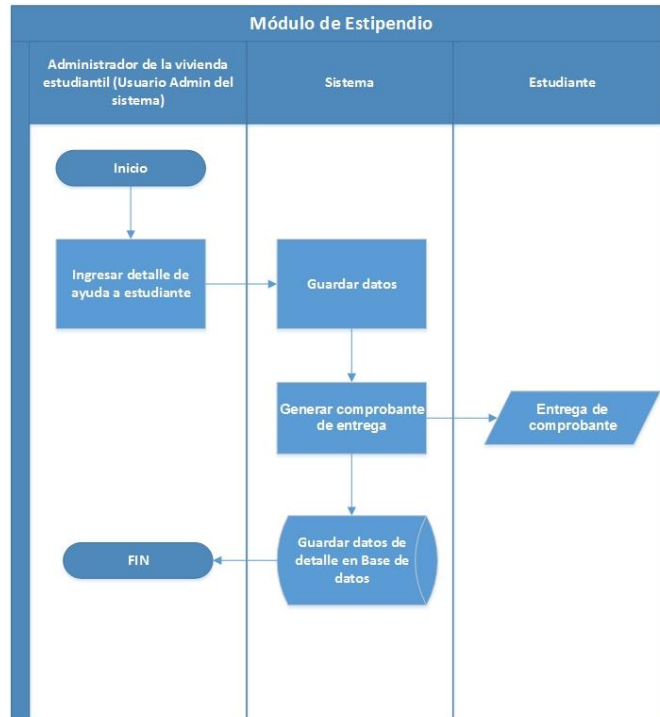
10.13 Anexo 12: Diagramas lenguaje unificado de modelado (UML) para el diseño lógico del sistema de información web para la Vivienda Estudiantil

Diagramas de flujo

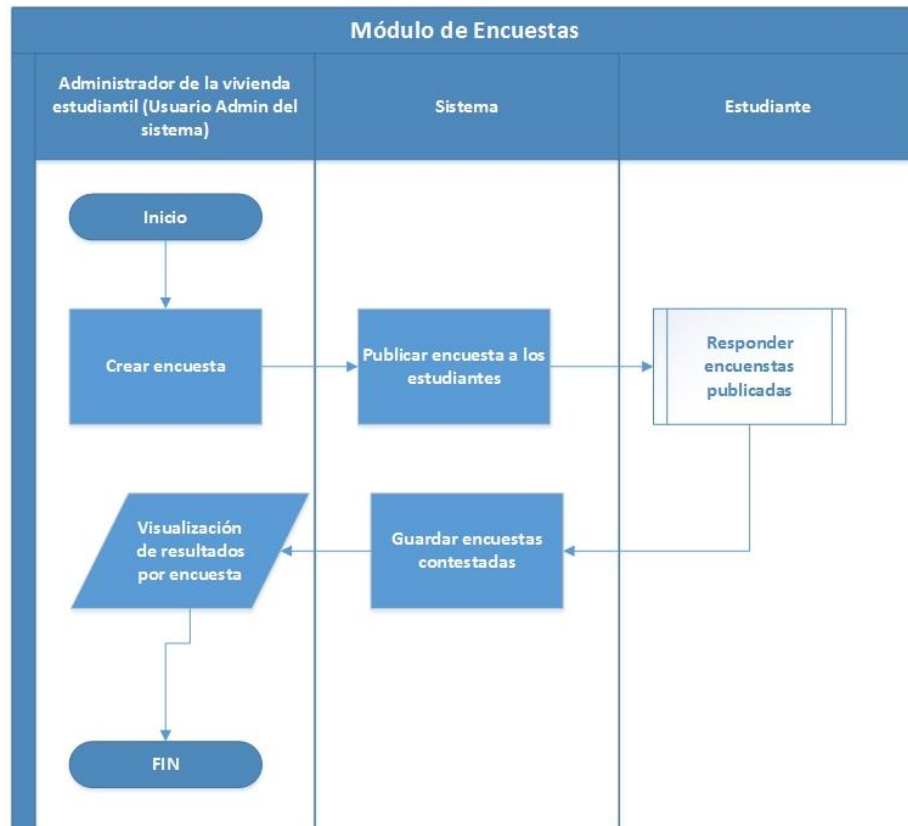
Módulo de rol de aseo



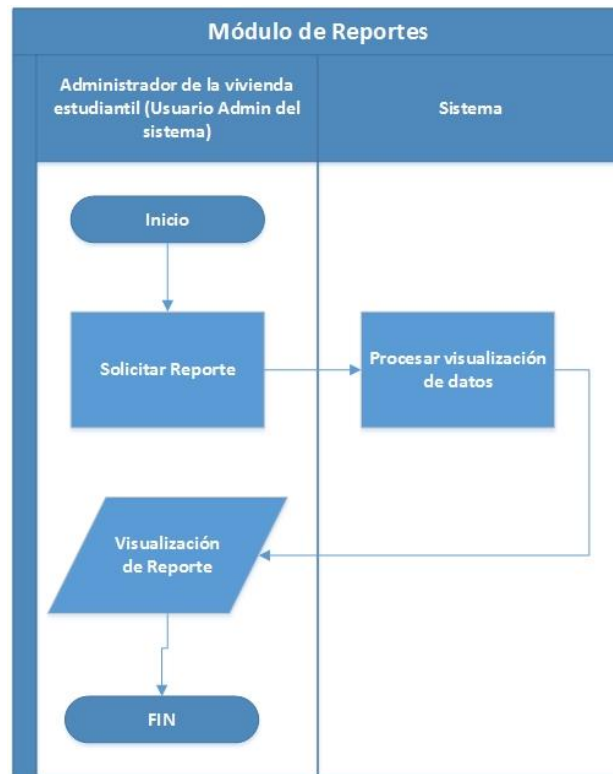
Módulo control de estipendio



Módulo de encuestas

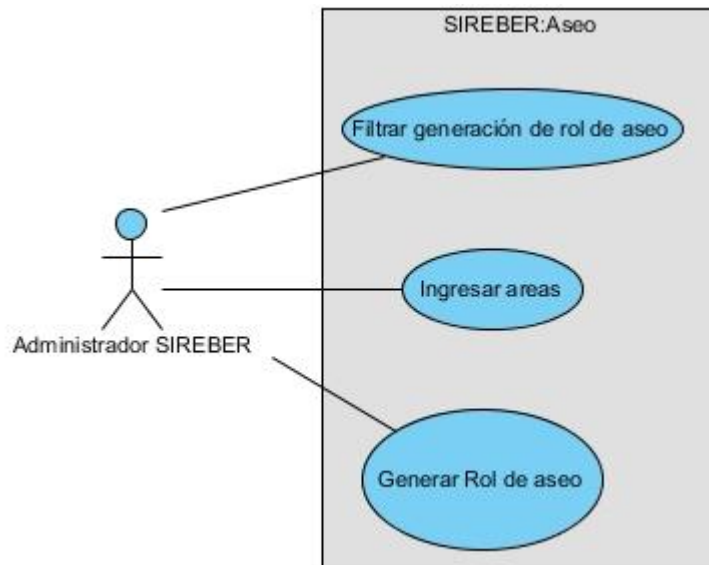


Módulo de reportes

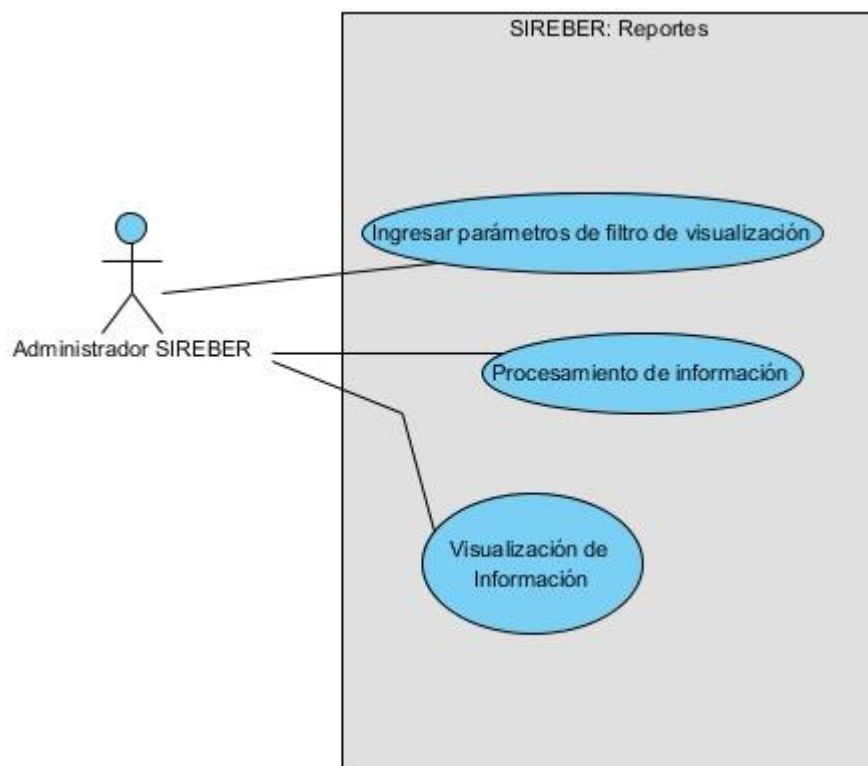


Diagramas de caso de uso

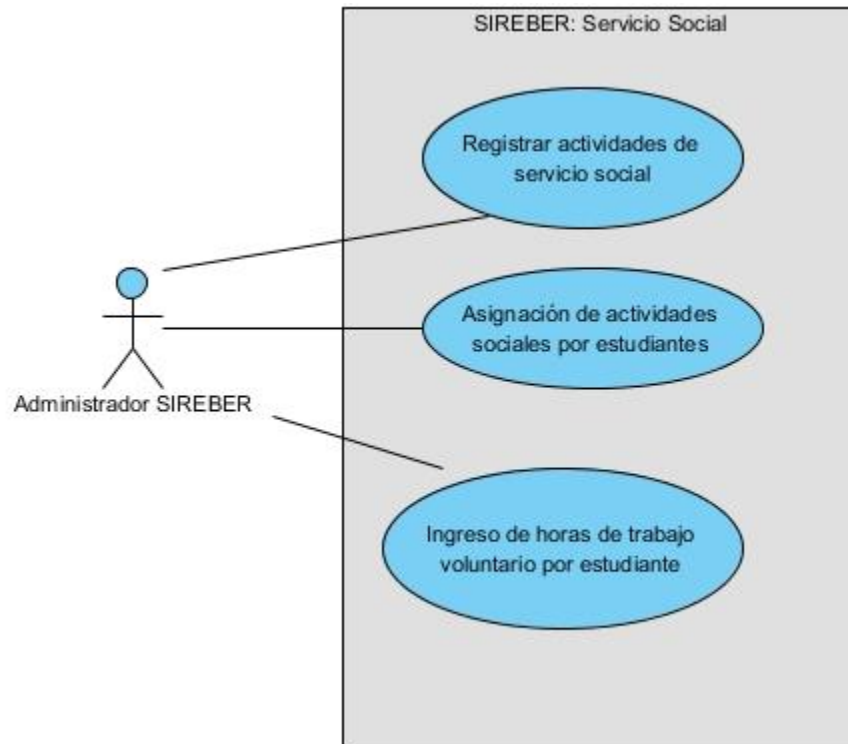
Módulo rol de aseo:



Módulo de reportes:



Módulo de actividad social



Diagramas de actividad

Diagrama de actividad asociado a caso de uso: Baja de artículos por cuarto y estudiantes, actor: Administrador.

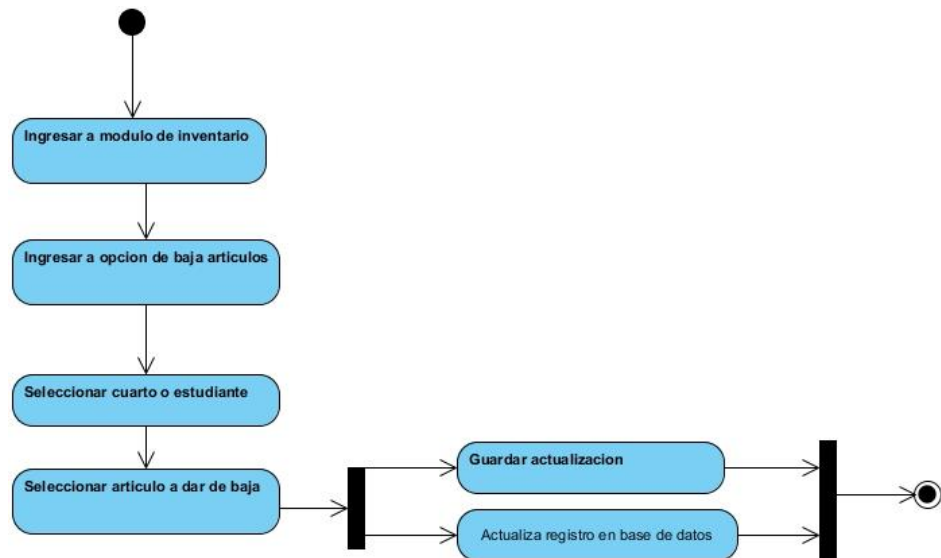


Diagrama de actividad asociado acaso de uso: Reporte de existencias y gastos generales, actor: Administrador.

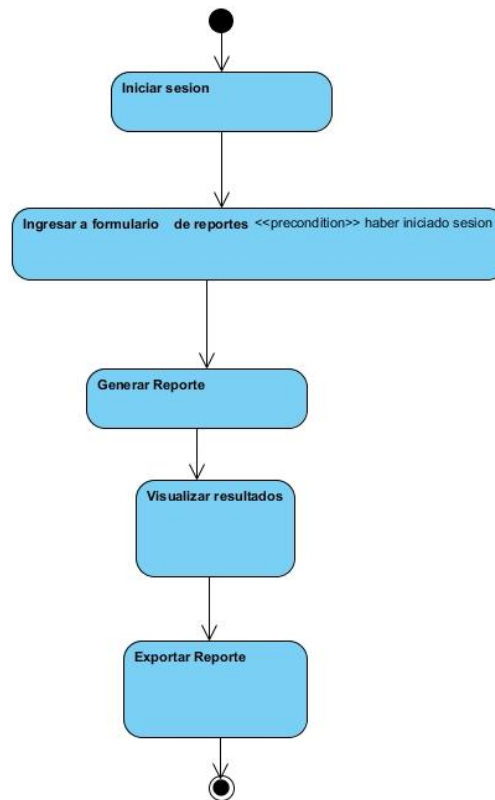


Diagrama de actividad asociado acaso de uso: Configurar entrega de comidas a estudiantes mediante carnet en los comedores, actor: Administrador.

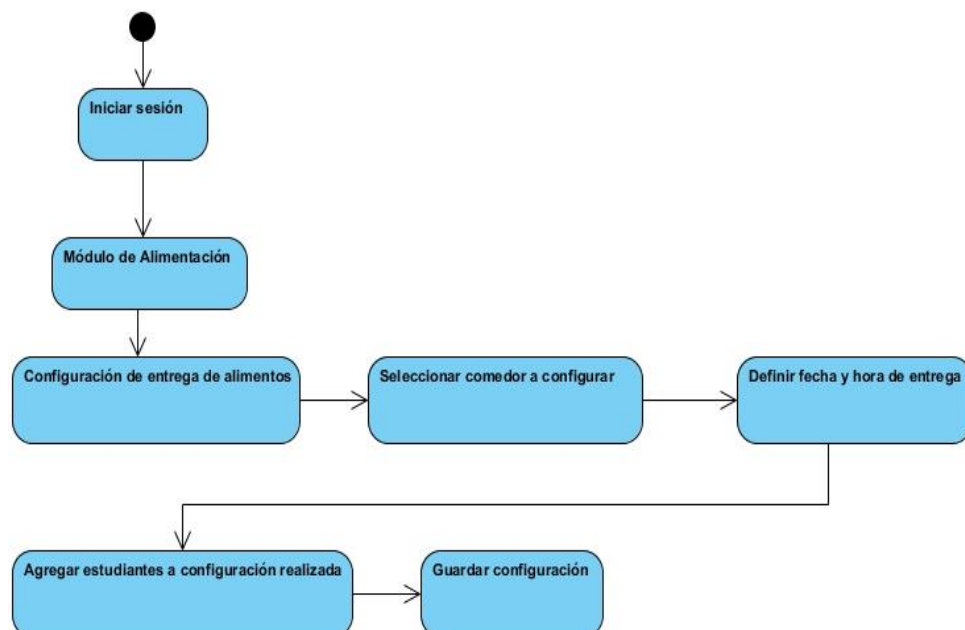


Diagrama de actividad asociado acaso de uso: Cargar menú de comida y su costos, actor: Administrador

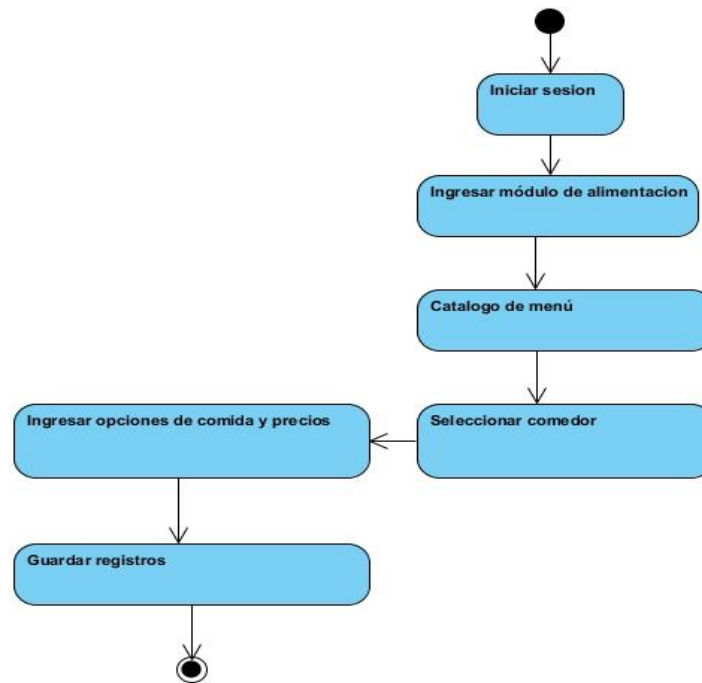


Diagrama de actividad asociado acaso de uso: Crear actividades de servicio social, actor: Administrador

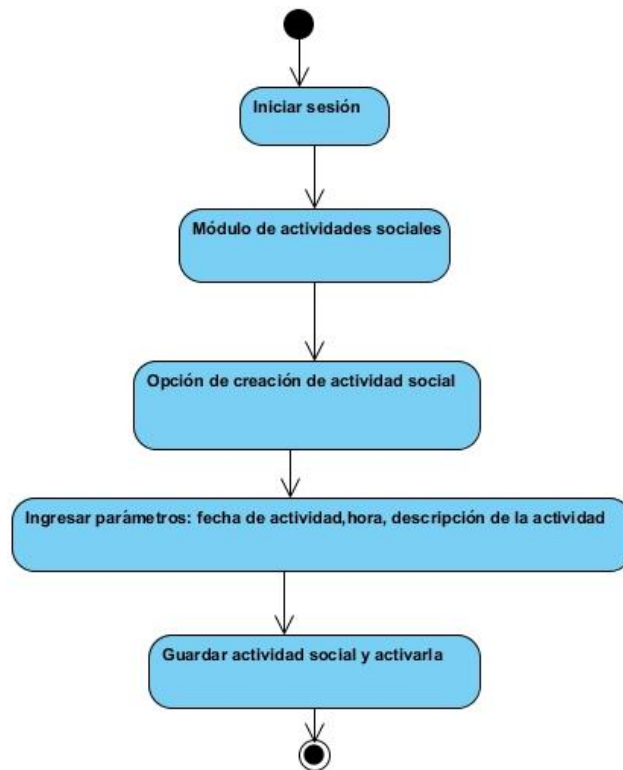


Diagrama de actividad asociado acaso de uso: Asignación de actividades sociales por estudiantes, actor: Administrador

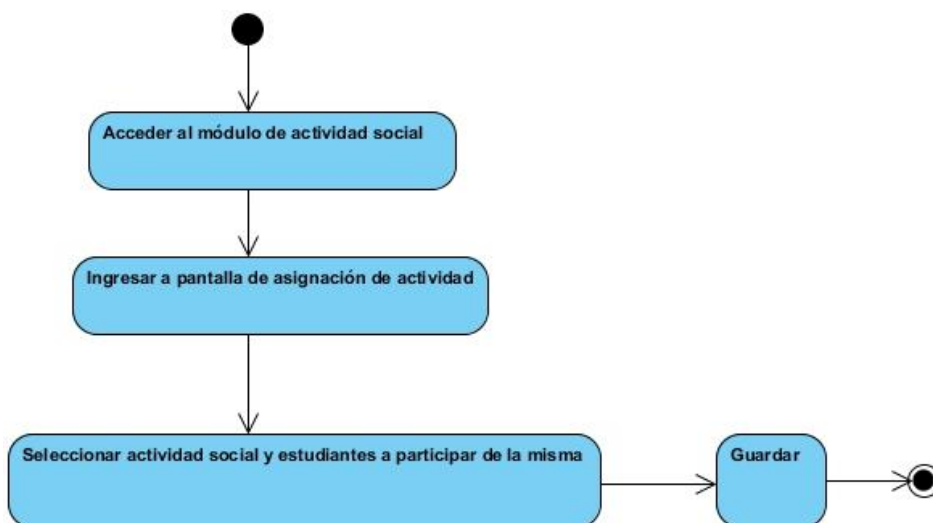


Diagrama de actividad asociado acaso de uso: Ingreso de horas de trabajo voluntario por estudiante, actor: Administrador

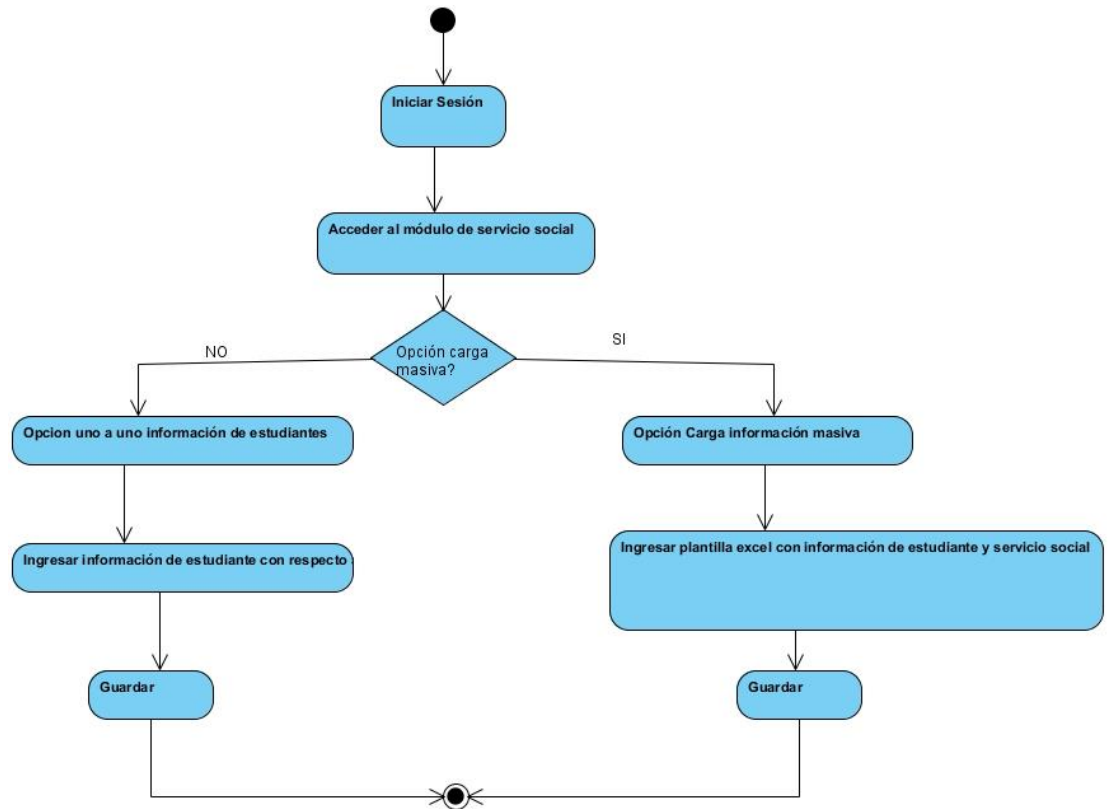


Diagrama de actividad asociado acaso de uso: Creación y activación de encuestas, actor: Administrador

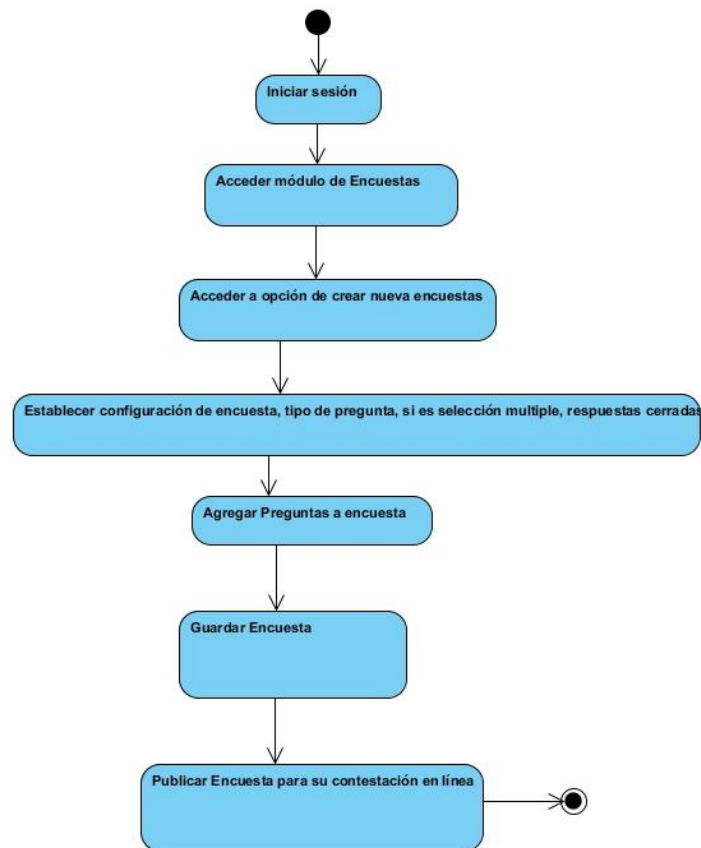


Diagrama de actividad asociado acaso de uso: Contestar encuestas, actor: Estudiante

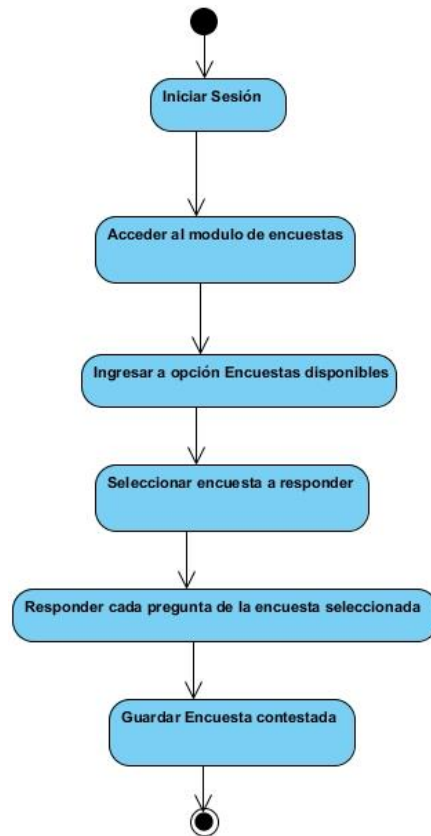


Diagrama de actividad asociado acaso de uso: Visualización de resultados y encuestas en general, actor: Administrador

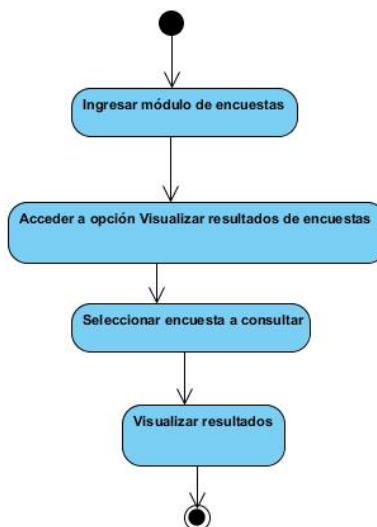
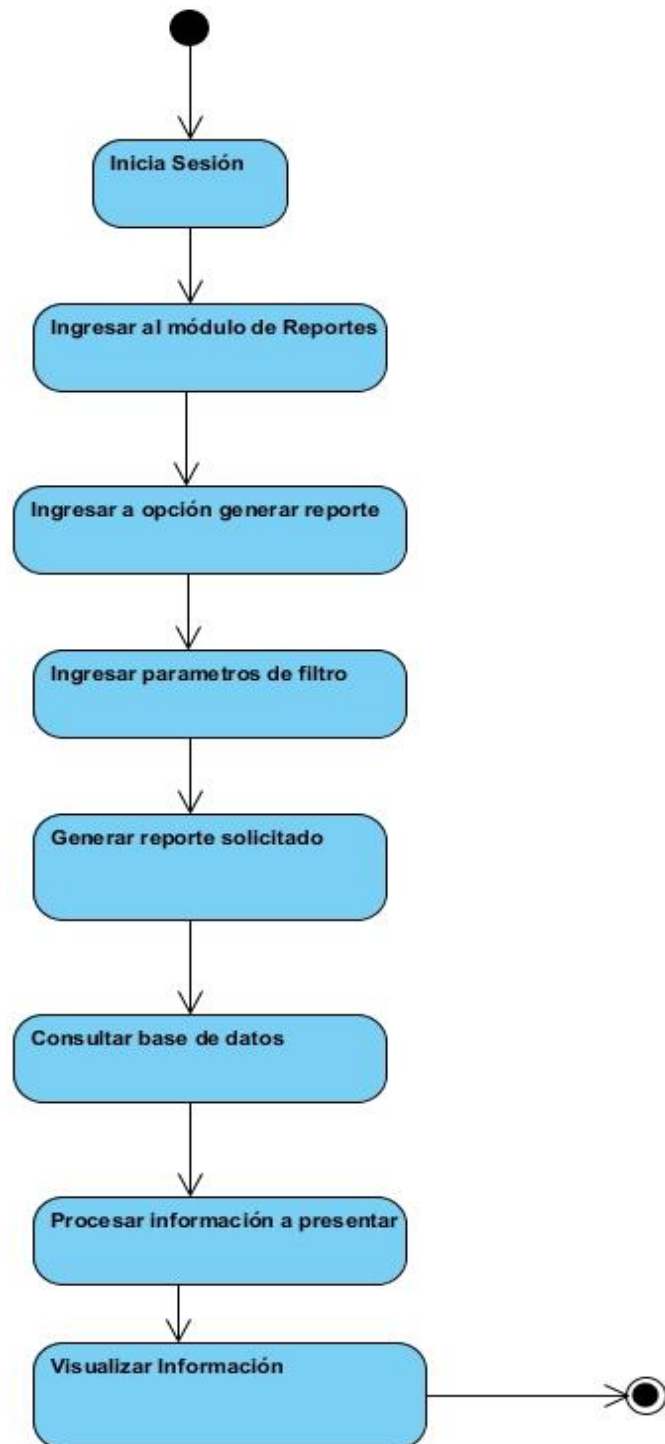


Diagrama de actividad asociado acaso de uso: Visualización de Información,
actor: Administrador



Diagramas de secuencia

Diagrama de secuencia asociado a caso de uso: Registro de enseres / inventario nuevo en bodega, actor: Administrador

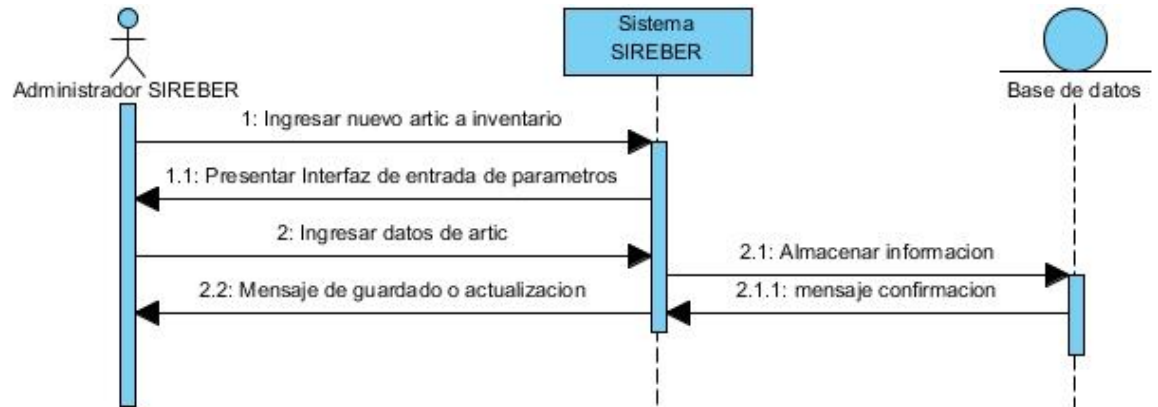


Diagrama de secuencia asociado a caso de uso: Reporte de existencias y gastos generales, actor: Administrador

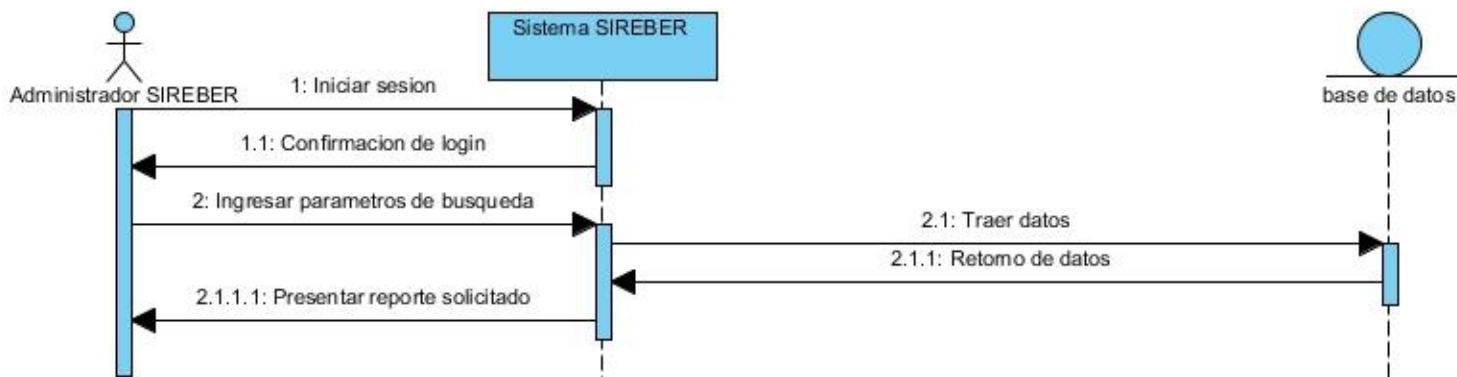


Diagrama de secuencia asociado a caso de uso: Crear actividades de servicio social, actor: Administrador

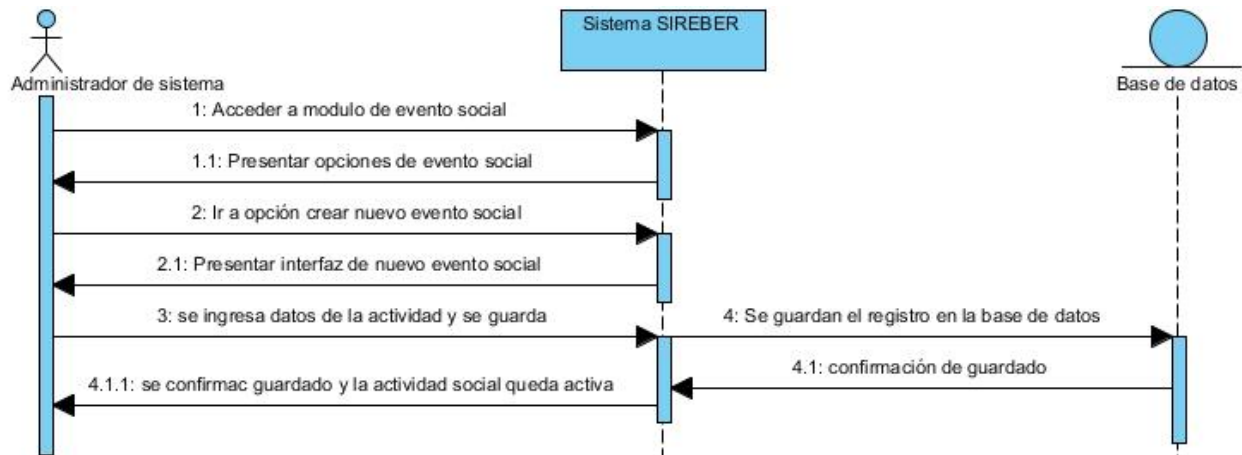


Diagrama de secuencia asociado a caso de uso: Ingreso de horas de trabajo voluntario por estudiante, actor: Administrador

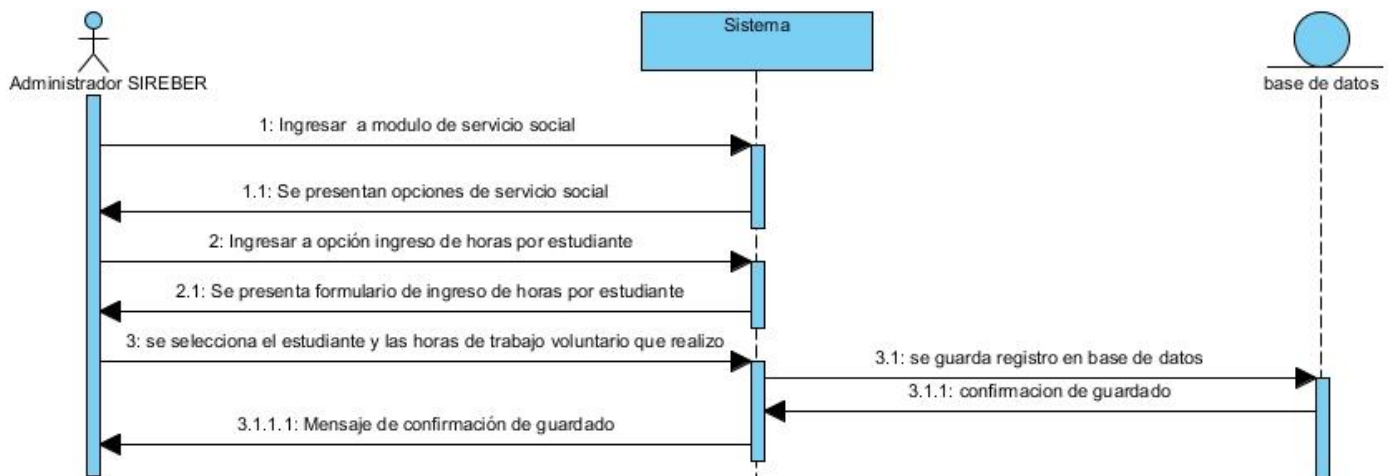


Diagrama de secuencia asociado a caso de uso: Visualización de resultados y encuestas en general, actor: Administrador.

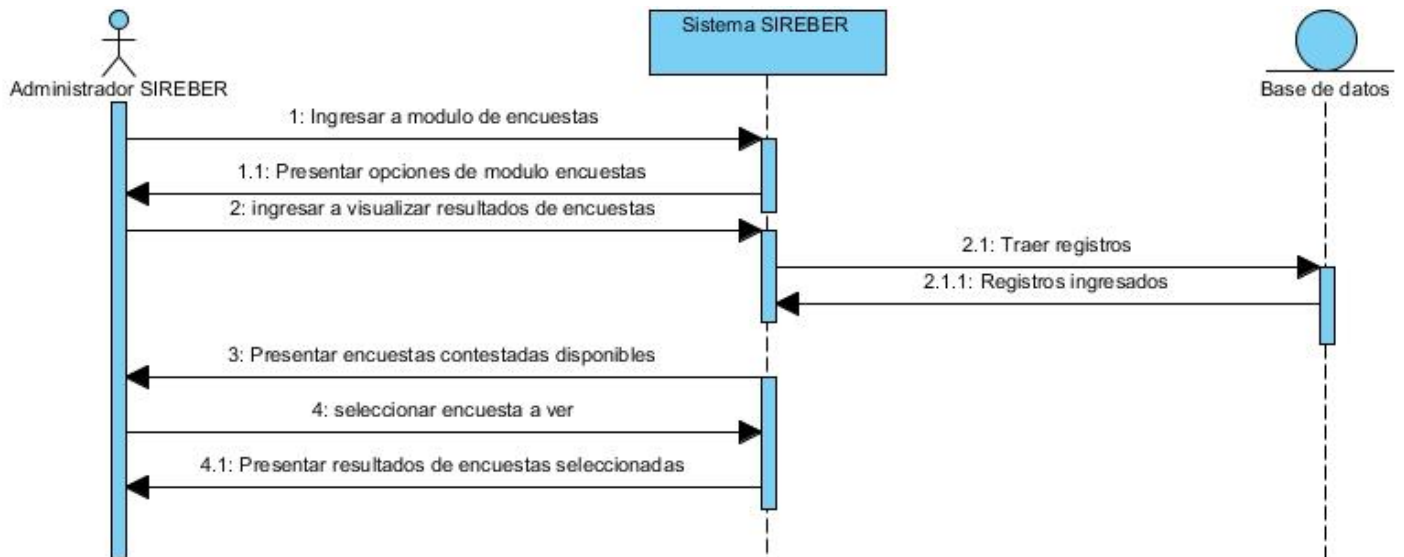


Diagrama de secuencia asociado a caso de uso Visualización de Información en reporteria, actor: Administrador

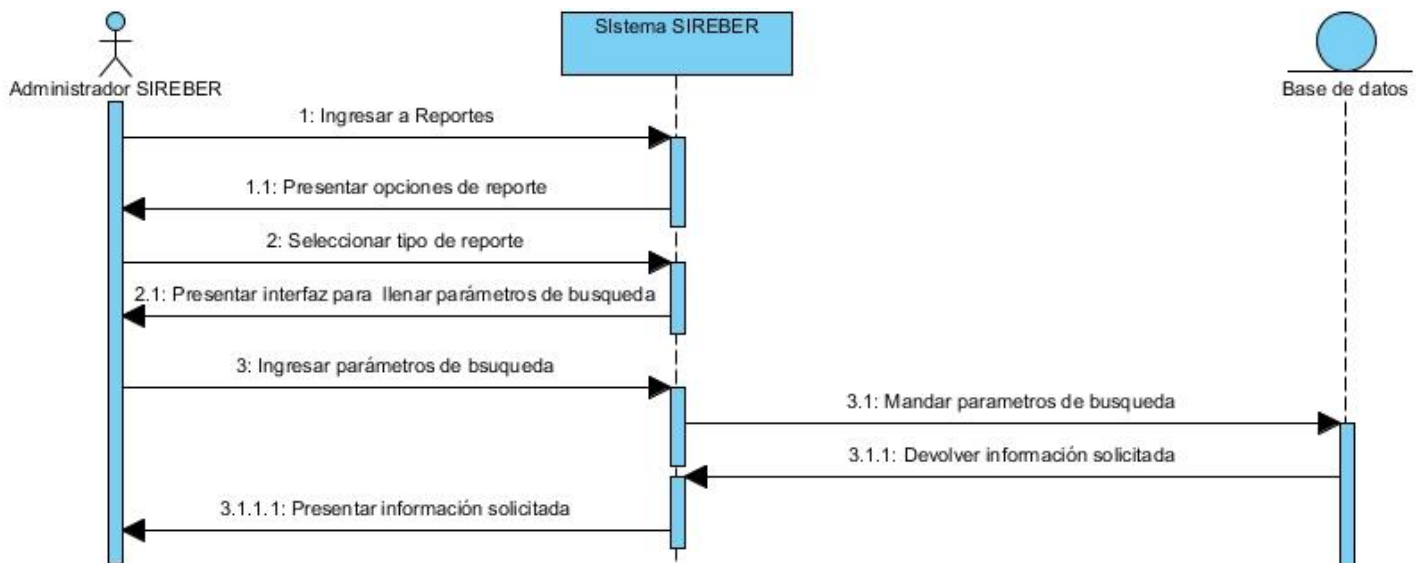
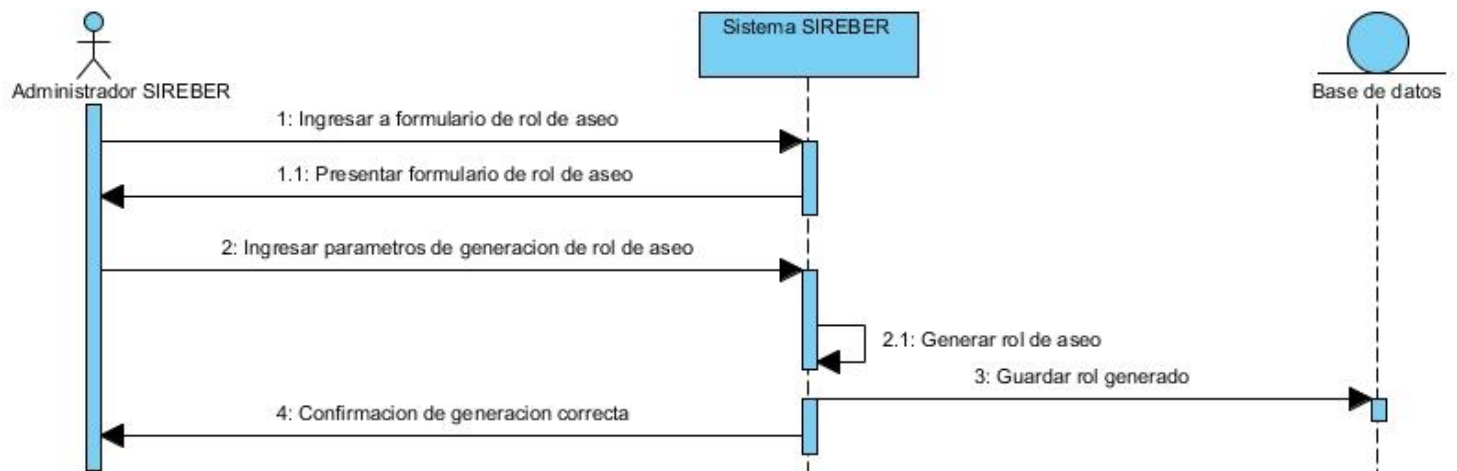


Diagrama de secuencia asociado a caso de uso Generar Rol de aseo, actor:
Administrador



10.14 Anexo 13: Load y performance Test

Load Test Summary

Test Run Information

Load test name	SireberLoadTest
Description	
Start time	28/11/2016 8:57:01 a. m.
End time	28/11/2016 9:02:01 a. m.
Warm-up duration	00:00:05
Duration	00:05:00
Controller	Local run
Number of agents	1
Run settings used	Run Settings1

Key Statistic: Top 5 Slowest Tests

Name	95% Test Time (sec)
ValidacionConsumoTest	0.00018
ObtenerConsumosTest	0.00016
LoginTest	0.000085

Overall Results

Max User Load	100
Tests/Sec	3,020
Tests Failed	0
Avg. Test Time (sec)	0.000026
Transactions/Sec	0
Avg. Transaction Time (sec)	0
Pages/Sec	-
Avg. Page Time (sec)	-
Requests/Sec	-
Requests Failed	-
Requests Cached Percentage	-
Avg. Response Time (sec)	-
Avg. Content Length (bytes)	-

Test Results

Name	Scenario	Total Tests	Failed Tests (% of total)	Avg. Test Time (sec)
ObtenerConsumosTest	ConsumoAlimenticioLoadTest	299,044	0 (0)	0.000030
ValidacionConsumoTest	ConsumoAlimenticioLoadTest	299,241	0 (0)	0.000028
LoginTest	ConsumoAlimenticioLoadTest	307,862	0 (0)	0.000019

Figura 29: Resultados load test

Run All Run... Playlist: All Tests	Request	Status	Total Time	Request Time	Request Bytes	Response Bytes
Passed Tests (3)						
✓ LoginTest 69 ms	✓ ▶ http://localhost:44300/	302 Found	0.019 sec	0.016 sec	54	145
✓ ObtenerConsumosTest 29 ms	✓ ▶ http://localhost:44300/Estudiantes/EstudianteCuarto/Index/	302 Found	0.005 sec	0.001 sec	0	186
✓ ValidacionConsumoTest 362 ms						

[TestMethod]

✓ | 0 references

public void ObtenerConsumosTest()

```
{
    // Arrange
    var db = new SireberEntities();
    var estudianteService = new EstudianteService(db);
    var horarioAlimenticioService = new HorarioAlimenticioService(db);
    var consumoAlimenticioService = new ConsumoAlimenticioService(db);
    var datatableParams = new DataTablesParam();

    var controller = new ConsumoAlimenticioController(estudianteService, consumoAlimenticioService, horarioAlimenticioService, consumoAlimenticioService);

    // Act
    var result = controller.ListarConsumoAlimenticio(new Mvc.Bootstrap.DataTables.DataTablesParam(), DateTime.Now);

    // Assert
    Assert.AreEqual(TaskStatus.RanToCompletion, result.Status);
}
```

```

[TestClass]
0 references
public class LoginControllerTest
{
    [TestMethod]
    0 references
    public void LoginTest()
    {
        // Arrange
        var loginController = new AccountController();

        var model = new LoginViewModel{
            Email = "william.palacios@yahoo.com",
            Password = "Sireber2015_",
            RememberMe = false
        };

        //Act
        var result = loginController.Login(model, "/");
    }
}

```

```

[TestMethod]
0 references
public void ValidacionConsumoTest()
{
    // Arrange
    var consumoService = new Mock<IConsumoAlimenticioService>();

    var db = new SireberEntities();
    var estudianteService = new EstudianteService(db);
    var horarioAlimenticioService = new HorarioAlimenticioService(db);
    var consumoAlimenticioService = new ConsumoAlimenticioService(db);
    var controller = new ConsumoAlimenticioController(estudianteService, consumoService.Object, horarioAlimenticioService, consumoAlimenticioService);
    int idEstudiante = 1;
    int idHorario = 3;

    // Act
    var result = controller.ValidarConsumo(idEstudiante, idHorario);

    // Assert
    Assert.AreEqual("{ error = Servicio alimenticio NO autorizado. }", result.Data.ToString());
}

[TestMethod]

```

10.15 Anexo 14: Test validación de información de salida y entrada

cantidad	Entero	Cantidad entregada del articulo
----------	--------	---------------------------------

Número de requisito	RF4
Nombre de requisito	Baja de artículos por cuarto y estudiantes
Tipo	Requisito
Fuente del requisito	Administrador vivienda estudiantil
Prioridad del requisito	Alta
Detalle	El sistema debe llevar el control de los artículos o enseres se dan de baja en un cuarto o a un estudiante. Ya sea porque el artículo se gastó o se dañó.

Parámetros de entrada:

Nombre del parámetro	Tipo de dato	Observación
IdArticulo	Entero	Id único del artículo a asignar
CódigoArticulo	String	Código que identifica al tipo de artículo de bodega que está en inventario
IdEstudiante	Número entero	Id único del estudiante
Idcuarto	Número entero	Id único del cuarto
cantidad	Entero	Cantidad entregada del artículo

Parámetro de salida:

Mensaje de confirmación de información actualizada

Mensaje de confirmación o negación de entrega de alimentos

Número de requisito	RF9
Nombre de requisito	Registro de histórico de entrega de comidas y costos
Tipo	Requisito
Fuente del requisito	Administrador vivienda estudiantil
Prioridad del requisito	Alta
Detalle	El sistema guarda las comidas entregadas cada mes a los estudiantes, se ingresan al sistema también la cantidad de alimentos retirados y el costo monetario.

Módulo de Servicio Social:

Número de requisito	RF11
Nombre de requisito	Registrar actividades de servicio social
Tipo	Requisito
Fuente del requisito	Administrador vivienda estudiantil
Prioridad del requisito	Media
Detalle	Se registran todas las actividades de servicio social activas, pueden ser limpieza de áreas verdes en la Universidad Nacional de Ingeniería, jornada de alfabetización o la actividad social que organice la dirección de bienestar estudiantil.

Módulo de Alimentación:

Número de requisito	RF6
Nombre de requisito	Registro de Recintos universitarios y comedores disponibles
Tipo	Requisito
Fuente del requisito	Administrador vivienda estudiantil
Prioridad del requisito	Alta
Detalle	Se ingresa en una tabla de catálogo los recintos universitarios y comedores disponibles para la entrega de comidas dentro de la UNI.

Parámetros de entrada:

Nombre del parámetro	Tipo de dato	Observación
IdUniversidad	Entero	Numero único que identifica a la universidad.
Nombre de universidad	String	Nombre propio de la Universidad, Ej: Universidad Nacional de Ingeniería.
Nombre del recinto	String	Nombre del recinto Universitario, una universidad puede tener varios recintos.
Idcomedor	Entero	Id único que identifica al comedor asignado a la universidad correspondiente

Número de requisito	RF12
Nombre de requisito	Asignación de actividades sociales por estudiantes.
Tipo	Requisito
Fuente del requisito	Administrador vivienda estudiantil
Prioridad del requisito	Media
Detalle	Se le asigna a cada estudiante actividades de servicio social, es decir que el usuario selecciona la actividad social, posteriormente selecciona a los estudiantes a añadir a la actividad, luego el usuario los guarda y se queda vinculado los estudiantes a la actividad social a realizar. En el sistema se pueden consultar y visualizar las listas de actividades culminadas y por completar.

Parámetros de entrada:

Nombre del parámetro	Tipo de dato	Observación
IdActividad	Entero	Tipo de dato que solo admite números enteros, Ej.: 1. El IdActividad es el identificador único de cada actividad.
NombreActividad	String	El tipo de dato String permite cadena de caracteres, letras o números.
NumeroCarnet	String	El tipo de dato String permite cadena de caracteres, letras o números. Numero único de carnet que identifica a cada estudiante.

Cuarto	Entero	Id único del cuarto
Cantidad	Entero	Cantidad entregada del artículo

Número de requisito	RF3
Nombre de requisito	Bitácora de gastos por estudiantes y por cuarto
Tipo	Requisito
Fuente del requisito	Administrador vivienda estudiantil
Prioridad del requisito	Alta
Detalle	Se debe hacer un registro de cuánto dinero se invierte en un estudiante y también por cuarto. Se debe guardar un histórico de los artículos asignados y sumar en un reporte final la cantidad de dinero total gastada en cada estudiante o cuarto. Se puede verificar en una lista general de nombres de los estudiantes o consultar uno a uno.

Parámetros de entrada:

Nombre del parámetro	Tipo de dato	Observación
IdArticulo	Entero	Id único del artículo a asignar
CódigoArticulo	String	Código que identifica al tipo de artículo de bodega que está en inventario
IdEstudiante	Numero entero	Id único del estudiante
Idcuarto	Numero entero	Id único del cuarto

descripción	String	Breve descripción del producto a guardar
Unidades guardadas	Numero entero	Cantidad inicial en existencia registrada
Total	Entero	Suma total guardada

Parámetro de salida:

Mensaje de confirmación de guardado exitoso

Número de requisito	RF2
Nombre de requisito	Asignación de artículos por estudiantes y por cuarto
Tipo	Requisito
Fuente del requisito	Administrador vivienda estudiantil
Prioridad del requisito	Alta
Detalle	Se debe implementar una pantalla donde se pueda asignar artículos de la bodega a un estudiante o a un cuarto, la pantalla debe generar una plantilla de entrega y guardar el registro de los articulo asignados en la base de datos, por defecto los cuartos tienen ya algunos enseres asignados.